Novel formulations and modeling enhancements for the dynamic berth allocation problem / Kramer, Arthur; Lalla-Ruiz, Eduardo; Iori, Manuel; Voß, Stefan. - In: EUROPEAN JOURNAL OF OPERATIONAL RESEARCH. - ISSN 0377-2217. - 278:1(2019), pp. 170-185. [10.1016/j.ejor.2019.03.036]

Novel formulations and modeling enhancements for the dynamic berth allocation problem

Arthur Kramer[a,], Eduardo Lalla-Ruiz[c,*], Manuel Iori[a], Stefan Voß[b]

[a]*Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia*
Via Amendola 2, Pad. Morselli, 42122 Reggio Emilia, Italy
[b]*Institute of Information Systems, University of Hamburg*
Von-Melle-Park 5, Room 3065 20146 Hamburg, Germany
[c]*Department of Industrial Engineering and Business Information Systems University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands*

Abstract

This paper addresses the well-known dynamic berth allocation problem (DBAP), which finds numerous applications at container terminals aiming to allocate and schedule incoming container vessels into berthing positions along the quay. Due to its impact on ports performance, having efficient DBAP formulations is of great importance, especially for determining optimal schedules in quick time as well as aiding managers and developers in the assessment of solution strategies and approximate approaches. In this work, we propose two novel formulations a time-indexed formulation and an arc-flow one, to efficiently tackle the DBAP. Additionally, to improve computational performance, we propose problem-based modeling enhancements and a variable-fixing procedure that allows to discard some variables by considering their reduced costs. By means of these contributions, we improve the models performance in those instances where the optimal solutions were already known, and we solve to optimality for the first time other instances from the literature.

Keywords: OR in maritime industry, Dynamic berth allocation problem, Novel formulations, Modeling enhancements

---

1. Introduction

The management of limited resources at maritime container terminals has a direct and relevant impact on their productivity and competitiveness. This holds, especially, in those cases where for geographical or monetary reasons the terminals are enforced to find different

---

[*] Corresponding author

Email addresses: `arthur.kramer@unimore.it` (Arthur Kramer), `e.a.lalla@utwente.nl` (Eduardo Lalla-Ruiz), `manuel.iori@unimore.it` (Manuel Iori), `stefan.voss@uni-hamburg.de` (Stefan Voß)

ways to expand their capacity. Thus, terminal managers require the use of efficient methods and approaches to efficiently exploit resources at maritime terminals. This involves the need for reliable and fast approaches for providing schedules within reasonable computational times, as well as having efficient mathematical models enabling the proper evaluation of those schedules by means of a given objective function. In this context, as indicated by Notteboom (2006), over 90% of the delayed vessel schedules are due to port access and terminal operations that, as pointed out by Steenken et al. (2004), directly involve the management of berths. Hence, it becomes essential for terminal operators and related practitioners to rely on efficient solution approaches in order to suitably manage the use of those limited and impacting resources such as berths.

The above issue leads to the definition of the well-known berth allocation problem (BAP), which seeks to assign and schedule incoming vessels arriving at the terminal into berthing positions with the aim of optimizing a given objective function (e.g. minimize vessels waiting time, maximum departure time, etc.). In this way, optimal berthing position and time for all vessels are provided, allowing planned berthing instructions while efficiently using the quay space. Different variants of the BAP have been proposed (see e.g. Bierwirth and Meisel 2010, 2015). Among them, the most referenced and known one is the dynamic berth allocation problem (DBAP, Imai et al. 2001; Cordeau et al. 2005). The DBAP aims at allocating container vessels along the quay partitioned into berths while reducing the sum of vessels turnaround time. In contrast to the static case (where all the vessels are at the port at the beginning of the planning horizon), the DBAP considers that vessels arrive along the planning horizon, i.e., the term "dynamic" means that the vessels arrive at different times of the day, nevertheless all problem information is known in advance. Due to the difficulty of this problem, decision support approaches are necessary to provide proper solutions. This opens up the discussion on how and from which standpoint algorithmic techniques can contribute to enhancing the management of berthing resources in port environments. Based on the literature (e.g. Bierwirth and Meisel 2010; Stahlbock and Voß 2008; Lalla-Ruiz 2016) different ways for solving the DBAP can be indicated:

1. Heuristic and metaheuristic approaches. They allow decision makers to provide feasible solutions within reasonable computational times. The application of these methods is suggested in situations where fast solutions are requested such as in dynamic environments requiring replannning or in integrative and rich solution systems where the solution of a problem is an input for other problems.

2. Matheuristic approaches. They integrate both exact and metaheuristic techniques,

in such a way that some of the capabilities of those methods can be jointly exploited. The integration of exact approaches within metaheuristic or vice-versa can lead to higher computational times than heuristic methods, but may also lead to a better performance robustness and quality of the solutions in some applications.

3. Exact approaches. The main advantage of this type of approaches is to seek optimality by means of bounds at the cost of requiring important computational efforts as the size of the instance increases. Counting on well-defined and assessed mathematical formulations, as well as exact algorithms, permits researchers and practitioners to evaluate the performance of their heuristic and matheuristic methods, while increasing the expertise and knowledge on the given problem and technique.

In recent years, the research community has been predominantly proposing heuristic approaches for the DBAP. That provides an incentive to study and develop exact approaches and modeling enhancements in order to aid the evaluation of the heuristics performance. On the other hand, assessing and determining the best formulation considering the evolution of exact solvers permits accelerating the resolution time as well as obtaining additional insights regarding the problem itself. Therefore, in this work, we aim at proposing two different novel ways of modeling the DBAP, i.e., a time-indexed formulation and an arc-flow one. As a follow up of previous studies on this problem, see Buhrkal et al. (2011), our goal is to provide a detailed comparison between our formulations and the best one proposed in the literature so far, in order to determine their performance and their likely complementarity for tackling the different DBAP benchmark instances. Furthermore, this work also aims at proposing and assessing modeling enhancements and a reduced-cost based variable-fixing procedure. As discussed below, the results are meaningful as our new formulations enable a relevant time reduction as well as provide optimal solutions not yet reported for several large-size problem instances proposed in the related literature. In addition to that, in order to address more congested scenarios as well as study the performance of the modeling approaches, a new set of large-size problem instances is proposed.

The remainder of this paper is organized as follows. Section 2 reviews the related literature putting an emphasis on mathematical models. In Section 3, the DBAP is described. Next, the currently best formulations for this problem as well as those proposed in this work are presented in Section 4. Their computational assessment, as well as a detailed comparison, is reported in Section 5. Finally, Section 6 presents the main conclusions of this work and proposes possible future research directions.

2. Related works

The dynamic berth allocation problem (DBAP) was initially proposed by Imai et al. (2001) with the goal of scheduling and allocating vessels along a discrete quay partitioned into berths. Due to its practical and relevant application domain, this problem has attracted a considerable and increasing attention from the research community as well as practitioners. Cordeau et al. (2005) reformulated the problem as a multi-depot vehicle routing problem with time-windows (MDVRP-TW) and proposed a tabu search for solving it. In this way, time-window constraints related to contractual agreements between shipping companies and container terminals could be incorporated. Their computational experiments were conducted on scenarios from the container terminal of Gioia Tauro (Italy) and the results indicated that the MDVRP-TW formulation was not able to solve small and medium-sized problem instances within the time limit. Christensen and Holst (2008) proposed a generalized set-partitioning problem formulation (GSPP) that is described in detail in Section 4 below. Later, in the work of Buhrkal et al. (2011), all the existing formulations proposed for the DBAP were extensively assessed. The authors indicated that the GSPP formulation clearly outperforms the other formulations in terms of linear bounds and computational time for the problem instances proposed in Cordeau et al. (2005). Nevertheless, Lalla-Ruiz et al. (2012) studied the GSPP performance on new instances and indicated that, under the computer and general purpose solver version used at that time, the formulation required high-amounts of memory, possibly leading to memory fault problems. Lalla-Ruiz and Voß (2016b) proposed a matheuristic decomposition approach in order to reduce the size of the problems and allowing to tackle them by means of the GSPP formulation. Recently, Nishi et al. (2016) proposed a new dynamic programming based matheuristic together with new instances to capture congested and larger scenarios. The authors used the GSPP formulation that, thanks to the progress of computers' memory and processors as well as software, allowed to avoid memory problems.

With regards to approximate approaches, the DBAP has attracted remarkable attention. We focus here on the most recent approaches. De Oliveira et al. (2012) proposed a clustering search with simulated annealing and Ting et al. (2014) proposed a particle swarm optimization approach. For testing their approaches both works only used the instances provided in Cordeau et al. (2005). Lalla-Ruiz et al. (2016) proposed a cooperative decentralized search and provided a comparison with De Oliveira et al. (2012) and Ting et al. (2014), indicating a relevant time and performance improvement. Mauri et al. (2016) proposed an adaptive large neighborhood search and tested it on all the state-of-the-art instances. All the mentioned metaheuristic approaches reported high-quality solutions in reasonable computational times. Nevertheless, although they were able to provide the optimal solution values for the largest instances proposed by Cordeau et al. (2005), they were not able to evaluate the quality of their approach for the instances of Lalla-Ruiz et al. (2012) as the optimal solutions remained unknown.

3. Problem description

In the DBAP, we are given a set $N = \{1, \ldots, n\}$ of vessels to be allocated within a quay that is divided into a set $M = \{1, \ldots, m\}$ of berths. Each vessel $i \in N$ is available to be served in a given time-window $[t_i, t'_i]$, where t_i and t'_i represent its arrival and departure time, respectively. Similarly, each berth $k \in M$ is available to serve vessels in a restricted period $[s_k, e_k]$. Furthermore, each vessel i has an associated handling time ρ_{ik} that depends on its assigned berth $k \in M$, and an input priority value p_i. The objective function of the DBAP is to minimize the total weighted flow time to serve incoming vessels, that is, the time elapsed between the vessels' arrival at the terminal and the completion of their associated operations multiplied by their priority values. Note that once a vessel has started to be served by a berth, its processing cannot be interrupted and restarted later on in the same or in another berth (i.e, preemption is not allowed).

Figure 1 presents an example of a DBAP solution. In the figure, a plan for six vessels within three berths is shown. The rectangles represent the vessels and their handling time. Inside each rectangle, we report the service priority of each vessel (p_i). The time-windows of the vessels are represented by the lines at the bottom of the figure. In this case, for example, vessel 1 arrives at time step 3 and should be served until time step 12. Moreover, the time-window of each berth is limited by the non-hatched areas. The vessels' handling times are reported in Table 1, those times depend on the assigned berth. Namely, for instance, if vessel 1 is assigned to berth 1, its handling time would be equal to 7, which is shorter than the handling time of 8 units that it would have at berth 2.

As indicated above, the objective value of a DBAP solution is the total weighted service time of the incoming container vessels. In this example, the weighted service times of the six vessels are calculated as follows: vessel 1 $= (10-3)\cdot(1) = 7$, vessel 2 $= (4-1)\cdot(3) = 9$, vessel 3 $= (6-2)\cdot(6) = 24$, vessel 4 $= (10-4)\cdot(4) = 24$, vessel 5 $= (11-2)\cdot(2) = 18$, and vessel 6 $= (13-11)\cdot(1) = 2$. Therefore, the objective function value of this solution is: $7 + 9 + 24 + 24 + 18 + 2 = 84$.

Table 1: Example of vessels handling times and priority values

Vessel	Handling times ρ_{ik}			Priority value p_i
	Berth 1	Berth 2	Berth 3	
1	7	8	6	1
2	2	3	4	3
3	5	5	4	6
4	8	6	5	4
5	9	8	5	2
6	4	2	5	1

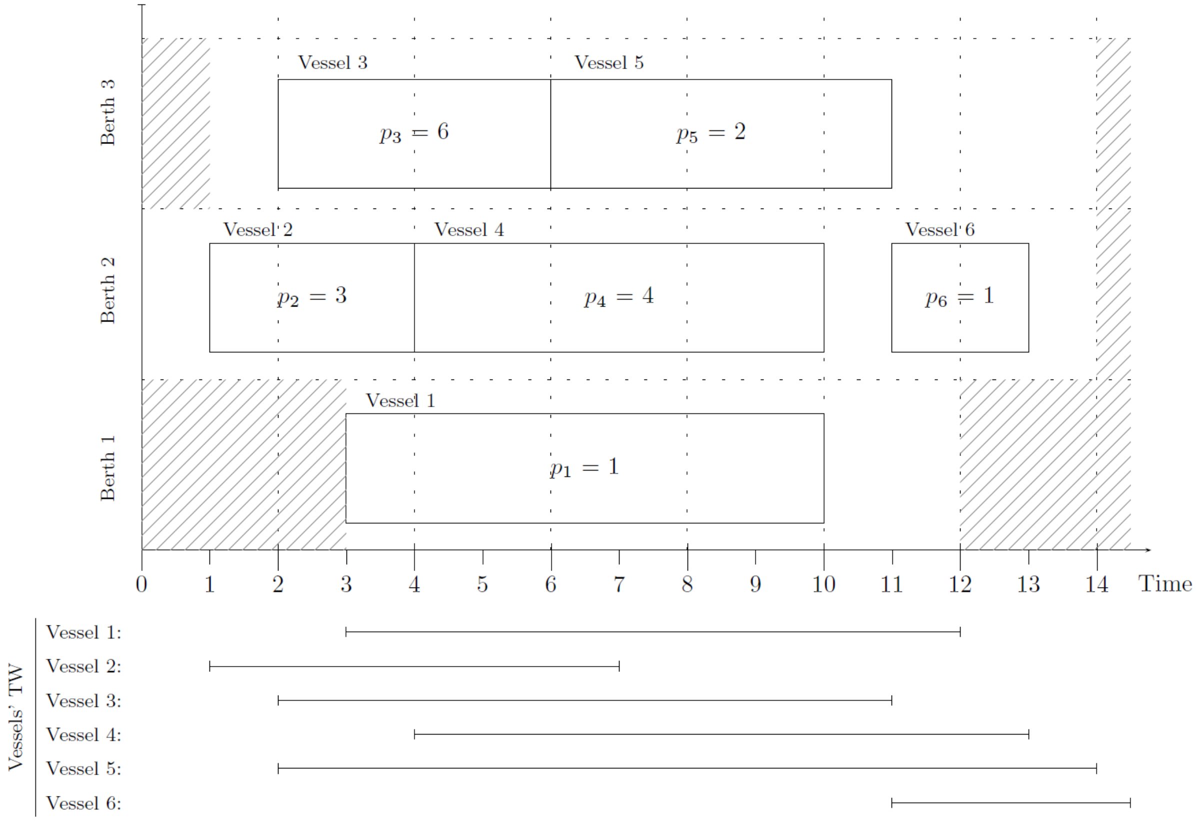

Figure 1: Example of a DBAP solution with six vessels and three berths. Hatched areas represent berth unavailability due to input time windows

4. Mathematical formulations for the DBAP

This section includes the current most efficient mathematical model for the DBAP according to the computational tests in Buhrkal et al. (2011), and the two novel models proposed in this work, namely, the time-indexed formulation and the arc-flow one.

4.1. Generalized set-partitioning problem formulation

The generalized set-partitioning problem (GSPP) formulation for the DBAP was proposed by Christensen and Holst (2008). In the GSPP formulation, a column represents a feasible assignment of a vessel to a berth at a certain time. The set of columns is denoted by Ω. Two matrices A and B are defined, both containing $|\Omega|$ columns. Matrix $A = (A_{i\omega})$ contains a row for each vessel, and $A_{i\omega} = 1$ if and only if column ω represents an assignment of vessel $i \in N$. Each column of A contains exactly one non-zero element. Matrix $B = (B_{p\omega})$ contains a row per (berth,time) position.

The rows of B are indexed by the set $P = \{1, 2, ..., K\}$ with $K = \sum_{k \in M}(e_k - s_k)$. The entry $B_{p\omega}$ is equal to 1, if and only if, position $p \in P$ is contained in the assignment that column ω represents. The cost c_ω of any column $\omega \in \Omega$ is the service time of the respective position assignment which are multiplied by the priority factor p_i. A binary variable x_ω is equal to 1 if column ω is used in the solution, and 0 otherwise. With these definitions the GSPP formulation for the DBAP is stated as follows.

$$(GSPP) \min \sum_{w \in \Omega} c_w x_w \tag{1}$$

subject to

$$\sum_{w \in \Omega} A_{iw} x_w = 1 \qquad i \in N, \tag{2}$$

$$\sum_{w \in \Omega} B_{pw} x_w \leq 1 \qquad p \in P, \tag{3}$$

$$x_w \in \{0, 1\} \qquad w \in \Omega. \tag{4}$$

The objective function (1) minimizes the weighted flow time of the vessels. Constraints (2) ensure that all vessels are served. Constraints (3) guarantee that at a time interval, in a berth, at most one vessel is served. Constraints (4) define the variables domain. This model contains $\mathcal{O}(nK)$ variables and $\mathcal{O}(n + K)$ constraints.

4.2. Time-indexed formulation

The time-indexed (TI) formulation considers the DBAP as an unrelated parallel machine scheduling problem with release dates and deadlines to minimize the total weighted flow time. In addition, it considers machine availability and job-machine incompatibilities. The TI formulation is an adaptation of the one originally proposed by Sousa and Wolsey (1992) for single machine scheduling problems. Let us define $u_{ik} = \min\{t'_i, e_k\}$ and $l_{ik} =$

$\max\{t_i, s_k\}, \forall i \in N, k \in M$. The TI formulation is then:

$$\text{(TI)} \quad \min \sum_{i\in N}\sum_{k\in M}\sum_{t=l_{ik}}^{u_{ik}-\rho_{ik}} p_i x_{ikt}(t+\rho_{ik}-t_i) \tag{5}$$

subject to

$$\sum_{k\in M}\sum_{t=l_{ik}}^{u_{ik}-\rho_{ik}} x_{ikt} = 1 \qquad i \in N, \tag{6}$$

$$\sum_{i\in N}\sum_{s=\max\{t_i,s_k,t+1-\rho_{ik}\}}^{\min\{t,t'_i-\rho_{ik},e_k-\rho_{ik}\}} x_{iks} \leq 1 \qquad k \in M, t = s_k, \ldots, e_k - 1, \tag{7}$$

$$x_{ikt} \in \{0,1\} \qquad i \in N, k \in M, t = l_{ik}, \ldots, u_{ik} - \rho_{ik} \tag{8}$$

where x_{ikt} is a binary variable taking value 1 if vessel i starts being served at time t by berth k, 0 otherwise. The objective function (5) seeks the minimization of the total weighted flow time of the vessels, where the flow time of a vessel i is defined by the difference between its service completion time and its arrival time at the port. Constraints (6) ensure that each vessel is served exactly once. Constraints (7) forbid overlapping among the vessels by imposing that at most 1 vessel is served by a berth at any time. Constraints (8) define the variables' domain. This model contains a pseudo-polynomial number of variables $\mathcal{O}(nK)$ and constraints $\mathcal{O}(n+K)$, a common characteristic of TI formulations.

4.3. Arc-flow formulation

Another new way of formulating the DBAP is by means of an arc-flow (AF) formulation. AF models represent problems by using flows on a capacitated network. The main idea is to obtain a one-unit flow from the origin to the sink node for each available resource. In our case, the berths are the resources. For each of them, the origin and the sink node can be seen as s_k and e_k, respectively, and the flow from origin to destination can be interpreted as a sequence of vessels served by the resource $k \in M$. AF formulations have been widely used to formulate different combinatorial optimization problems. In this sense, we address the reader to the works of Valério de Carvalho (1999) and Delorme et al. (2016).

Before introducing the proposed AF formulation, let us define $d_{kt}, \forall k \in M, t = s_k, \ldots, e_k - 1$ as a set of dummy variables necessary to allow the presence of idle times between the service of two consecutive vessels. Variable d_{kt} takes value 1 if in the time period from t to $t+1$ the berth k is idle. By using the previous set of variables x_{ikt} and the new variables d_{kt}, the DBAP can be formulated as follows:

$$\text{(AF)} \quad \min \sum_{i\in N} \sum_{k\in M} \sum_{t=l_{ik}}^{u_{ik}-\rho_{ik}} p_i x_{ikt}(t+\rho_{ik}-t_i) \tag{9}$$

subject to

$$\sum_{k\in M} \sum_{t=l_{ik}}^{u_{ik}-\rho_{ik}} x_{ikt} = 1 \qquad i \in N, \tag{10}$$

$$\sum_{i\in N} x_{ikt} - \sum_{i\in N} x_{i,k,t-\rho_{ik}} + d_{kt} - d_{k,t-1} = \begin{cases} 1, & \text{if } t = s_k \\ -1, & \text{if } t = e_k \\ 0, & \text{otherwise} \end{cases} \qquad {\scriptstyle k\in M, \atop t=s_k,\dots,e_k,} \tag{11}$$

$$x_{ikt} \in \{0,1\} \qquad {\scriptstyle i\in N, k\in M, \atop t=l_{ik},\dots,u_{ik}-\rho_{ik},} \tag{12}$$

$$0 \le d_{kt} \le 1 \qquad {\scriptstyle k\in M, \atop t=s_k,\dots,e_k-1.} \tag{13}$$

The objective function (9) and constraints (10) are equivalent to (5) and (6) in the TI formulation, respectively, whereas constraints (11) impose the flow conservation conditions. Like the TI formulation, the AF formulation too is characterized by a pseudo-polynomial number of variables, $\mathcal{O}(nK)$ binary and $\mathcal{O}(K)$ continuous, and constraints $\mathcal{O}(n+K)$. With the aim of illustrating this formulation, Figure 2 shows the AF solution for the example instance of Figure 1 and Table 1.

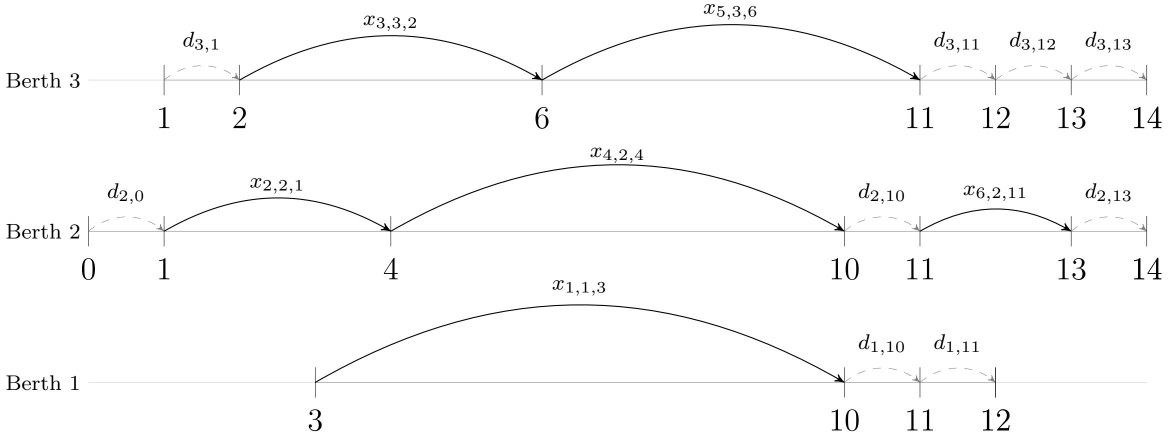

Figure 2: AF solution for the instance given in Figure 1 and Table 1

4.4. Equivalence between the mathematical models

All three formulations, GSPP (1)-(4), TI (5)-(8) and AF (9)-(13), model the DBAP by means of a pseudo-polynomial number of variables. Indeed, in all formulations, the main decision variable indicates the assignment of a vessel to a berth at a given starting time. It is not surprising thus that the three formulations are all equivalent to one another, i.e., they have the same continuous relaxation value. We skip a formal mathematical proof but give the reader a hint of this equivalence.

Let us first address the relation between GSPP and TI. In the GSPP model, set Ω contains all feasible assignments of a vessel to a (berth, time) position, and matrices $A^{n\times|\Omega|}$ and $B^{K\times|\Omega|}$, with $K=\sum_{k\in M}(e_k-s_k)$, indicate which vessel and (berth, time) positions, respectively, are associated with each variable x_ω, for $\omega\in\Omega$. Consider again the example of Table 1 and Figure 1. The first possible assignment is that of vessel 1 to berth 1 at time 3 (as both vessel and berth time windows start in 3). For TI, this is simply represented by variable x_{113}. For GSPP, this is instead represented by variable x_1 and its associated A entries satisfying $A_{11}=1$ and $A_{i1}=0$ for all $i\neq 1$, and B entries satisfying $B_{p1}=1$ for $p=1,2,\ldots,7$ (corresponding to times between 3 and 9) and $B_{p1}=0$ for $p=8,9$ (corresponding to times 10 and 11). The only entry taking the value 1 for A, A_{11}, indicates in constraint (2) that if x_1 is chosen, then vessel 1 has been assigned. The same result is obtained for TI by constraints (6) when x_{113} takes the value 1. The entries taking the value 1 in B indicate in constraints (3) that if x_1 is chosen then berth 1 is busy until time 10. The same result is obtained for TI by (7) when x_{113} takes the value 1. Extending this reasoning to all assignments, one can deduce that (2) can be directly mapped into (6), and (3) into (7). In addition to that, the domains of the variables are identical, as imposed by (4) and (8), so it follows that GSPP and TI are equivalent.

Concerning the relation between TI and AF, Valério de Carvalho (2002) proved that the two formulations are equivalent when applied to the cutting stock problem. Recently, a similar proof has been used by Kramer et al. (2018) to prove the equivalence of TI and AF for the problem of minimizing total weighted completion time on identical parallel machines. As the proposed TI and AF formulations for the DBAP rely on the same principles of the ones for the cutting stock and the parallel machine scheduling problem, we address the reader to these works for a proof of equivalence.

It follows that the three formulations are equivalent to one another. Despite this fact, their computational performances are remarkably different. This can be explained by a number of factors. Firstly, for GSPP the computation of the initial matrices A and B can be very memory consuming, and even prohibitive for very large instances. Secondly, it is known that commercial solvers are very sensitive to model details and initial conditions (see, e.g., Lodi and Tramontani 2013; Fischetti and Monaci 2014; Lalla-Ruiz and Voß 2016a). Changes in variables and constraints can thus deeply affect the model performance. Lastly, additional improvement techniques, like those discussed in the next two sections, may render even larger the difference between the performance of the models. All these behaviors can be observed in detail in our computational evaluation in Section 5.

4.5. Modeling improvements - grouping identical berths and vessels

In this section, we introduce new model improvements by considering some problem features like identical berth and vessels. We formalize the necessary conditions that berths and vessels have to comply in order to be considered identical. Through their proper identification and handling, we aim at reducing the number of variables and constraints of a given model. In the following, we define and indicate how to extract and integrate that information in a preprocessing step before starting to solve the DBAP.

In the DBAP, berths can be compared in terms of their features, i.e., time-windows and processing speed for serving incoming vessels. Thus, subsets of berths operating at the same service speed for the same vessels and sharing the same time-windows can be grouped as identical. This is formally defined by the below definitions.

Definition 1. Two berths $k \in M$ and $l \in M$, $k \neq l$, are considered identical if the following conditions are satisfied: $s_k = s_l$, $e_k = e_l$, and $\rho_{ik} = \rho_{il}$ for all $i \in N$.

Definition 2. A berth type is defined by those berth features that allow creating groups of identical berths. The set of berth types is indicated by M', such that $M' \subseteq M$, and $M' = M$ when no identical berths are detected. Additionally, for each berth type $k \in M'$, a resource amount a_k is defined as the number of berths of each berth type. Note that $\sum_{k \in M'} a_k = m$.

Similarly, it is also expected that incoming vessels might have the same features in terms of required service times for the same berth assignment, priorities, and time-windows, and can be consequently considered identical. Formally:

Definition 3. Two vessels $i \in N$ and $j \in N$, $i \neq j$, are identical if the following conditions are satisfied: $t_i = t_j$, $t'_i = t'_j$, $p_i = p_j$, and $\rho_{ik} = \rho_{jk} \forall k \in M$.

Definition 4. A vessel type is defined by those vessel features that allow to create groups of identical vessels. The set of vessel types is indicated by N', such that $N' \subseteq N$, and $N' = N$ when no identical vessels are detected. The total number of vessels of type $i \in N'$ is given by b_i, with $\sum_{i \in N'} b_i = n$.

Once sets M' and N' have been defined, one can easily modify the previous formulations to incorporate and make use of this information. Since including this reduction is similar to all formulations, in the following we only show it in the AF formulation (9)-(13).

$$(\mathrm{AF}_+) \quad \min \sum_{i \in N'} \sum_{k \in M'} \sum_{t=l_{ik}}^{u_{ik}-\rho_{ik}} p_i x_{ikt}(t + \rho_{ik} - t_i) \tag{14}$$

subject to

$$\sum_{k \in M'} \sum_{t=l_{ik}}^{u_{ik}-\rho_{ik}} x_{ikt} = b_i \qquad i \in N', \tag{15}$$

$$\sum_{i \in N'} x_{ikt} - \sum_{i \in N'} x_{i,k,t-\rho_{ik}} + d_{kt} - d_{k,t-1} = \begin{cases} a_k, & \text{if } t = s_k \\ -a_k, & \text{if } t = e_k \\ 0, & \text{otherwise} \end{cases} \qquad {\scriptstyle k \in M', \atop t = s_k, \ldots, e_k,} \tag{16}$$

$$x_{ikt} \in \{0, \ldots, \min\{a_k, b_i\}\} \qquad {\scriptstyle i \in N', k \in M', \atop t = l_{ik}, \ldots, u_{ik}-\rho_{ik},} \tag{17}$$

$$0 \leq d_{kt} \leq a_k \qquad {\scriptstyle k \in M', \atop t = s_k, \ldots, e_k - 1.} \tag{18}$$

Constraints (15) now take into account that a vessel type $i \in N'$ should be served b_i times and constraints (16) allow a berth type $k \in M'$ to serve at most a_k vessels simultaneously. In constraints (17), variables x_{ikt} are now integer and upper bounded by $\min\{a_k, b_i\}$ while variables d_{kt} are still continuous, but now upper bounded by a_k.

4.6. Modeling improvements - reduced-cost variable-fixing algorithm

This subsection presents a reduced-cost-based variable-fixing procedure aimed at enhancing the starting conditions of the optimization models. Variable-fixing strategies have been studied in Savelsbergh (1994) and approaches considering them have been applied to well-known combinatorial optimization problems, such as multiple-choice knapsack (Gao et al., 2017), machine scheduling (Ibaraki and Nakamura, 1994; Pessoa et al., 2010; Baptiste et al., 2010), and vehicle routing (Baldacci et al., 2008), among others.

Our method attempts to reduce the number of variables of a mathematical model by using information given by the optimal solution of the linear model relaxation and by a heuristic DBAP solution. For convenience, we denote an instance of our DBAP problem as P, its optimal solution as x^*, a feasible solution as x^{UB} with objective value of $z(x^{UB})$, and the linear relaxation of P as $LP(P)$ with an optimal solution x^{LP} and objective value of $z(x^{LP})$. Moreover, related to x^{LP} we denote the reduced costs corresponding to variables x_i^{LP} as $\bar{c}_i$. Bearing in mind such notation, we state that a variable x_i can be fixed to zero in the model if the following condition holds:

$$\bar{c}_i > z(x^{UB}) - z(x^{LP}) - 1. \tag{19}$$

Suppose indeed there is a non-basic variable x_i whose reduced cost $\bar{c}_i$ is higher than $z(x^{UB}) - z(x^{LP}) - 1$. Then, if x_i enters the basis with one unit, the current LP objective value $z(x^{LP})$ will increase by $\bar{c}_i$, thus obtaining $z(x^{LP}) + \bar{c}_i > z(x^{UB}) - 1$. Therefore, any integer solution containing variable x_i will have cost at least $\lceil z(x^{LP}) + \bar{c}_i \rceil \geq z(x^{UB})$. We can thus remove x_i from the model as we are only interested in solutions that could improve the current incumbent value $z(x^{UB})$. This condition is formalized as follows:

Proposition 1. A non-basic variable x_i can be removed from the model if its reduced cost $\bar{c}_i$ satisfies inequality (19).

Algorithm 1: Variable-fixing algorithm for the DBAP

```
1 $z(x^{LP}) \leftarrow$ Solve the LP relaxation
2 $z(x^{UB}) \leftarrow$ Obtain a valid upper bound by a given method
3 for ($\forall x_i^{LP} \in x^{LP}$) do
4     if ($\bar{c}_i > z(x^{UB}) - z(x^{LP}) - 1$) then
5         $x_i = 0$
6 Construct reduced problem $\overline{P}$
7 Solve $\overline{P}$ by means of a general purpose solver
```

Algorithm 1 describes the overall reduced-cost variable-fixing procedure. At step 1, the linear relaxation of a given DBAP instance is solved. A feasible solution is obtained through a heuristic procedure at step 2. The variable-fixing is applied at steps 3-5 by considering inequality (19). After that, a reduced problem $\overline{P}$ is obtained and then solved. This preprocessing procedure requires having tight bounds in order to have a certain impact on the solving performance. Thus, in our current work, we use the state-of-the-art heuristic technique by Lalla-Ruiz et al. (2016) to obtain high-quality $z(x^{UB})$ values.

5. Computational results

This section presents the computational experiments carried out for assessing the performance of the proposed formulations. The models were coded in C++ and solved on a computer equipped with an Intel i5 3.20 GHz and 16 GB of RAM running under Windows 10 operating system. The models were solved with IBM CPLEX 12.8, using a single thread, and a time limit of 2 hours. The method used for generating the upper bounds is the one provided in Lalla-Ruiz et al. (2016).

5.1. Benchmark instances

In this work, we use the problem instances proposed in the literature by Cordeau et al. (2005), Lalla-Ruiz et al. (2012) and Nishi et al. (2016). Among those proposed by Cordeau et al. (2005), we consider the large-sized ones, which contain 60 vessels and 13 berths. Those instances were generated by taking into account a statistical analysis of the traffic and berth allocation data at the maritime container terminal of Gioia Tauro (Italy) and were also studied in Cordeau et al. (2007). Moreover, we tackle the instances proposed by Lalla-Ruiz et al. (2012) that could not be solved to optimality by the computer used to conduct their computational experiments. This set contains 90 instances with up to 60 vessels and 7 berths. We have also used the recently proposed problem instances by Nishi et al. (2016) that consider more congested scenarios with up to 150 vessels and 15 berths. In addition to that, we created 20 new very large instances having up to 250 vessels and 20 berths. All instances are available at http://github.com/elalla/DBAP/tree/master/Instances_Kramer-Lalla-Ruiz-Iori-Vo/.

5.2. Computational experiments on the instances from the literature

In this section, we report and discuss the results obtained by means of the previously introduced formulations (see Section 4). Namely, we compare the performance of GSPP (i,e., (1)-(4)), TI (i.e., (5)-(8)) and AF (i.e., (9)-(13)) formulations. In addition, we assess the contributions of the improvements provided in Sections 4.5 and 4.6. Thus, in the tables, the models incorporating the improvements presented in Subsection 4.5 are tagged with a “ $_{+}$ ”, while those also considering the reduced-cost-based variable-fixing procedure presented in Section 4.6 are indicated by a “ $\overset{\text{rc}}{+}$ ”.

Table 2 shows a summary of the size of the models in terms of average number of variables (*cols*) and constraints (*rows*) reported in thousands. It is worth mentioning that the results shown for the reduced-cost variable-fixing methods represent the size of the reduced mixed integer linear programming formulation obtained after fixing the respective variables to zero. As can be seen in the results, the improvements proposed in this work enable relevant reductions of the model sizes. In this regard, it can be noticed that grouping similar berths and vessels does not always lead to a model size reduction. On the contrary, using the variable-fixing approach results in relevant reductions in all cases. For instance, for the largest instances, the standard models have more than 600 thousands variables on average, while this value can be reduced to nearly 200 thousands in some cases. It is also worth mentioning that AF needs more variables to model the problem than GSPP and TI. This fact is due to the use of the continuous dummy variables d_{ik}.

Table 2: Comparison of formulations' size in terms of variables (cols) and constraints (rows), in thousands

			Generalized set partitioning						Time-indexed						Arc-flow					
n	m	#inst	GSPP		GSPP_+		$\text{GSPP}^{\text{rc}}_+$		TI		TI_+		TI^{rc}_+		AF		AF_+		AF^{rc}_+	
			cols	rows	cols	rows	cols	rows	cols	rows	cols	rows	cols	rows	cols	rows	cols	rows	cols	rows
30	3	10	44.0	1.8	36.0	1.8	3.3	1.8	44.0	1.8	36.0	1.8	3.3	0.8	45.8	1.8	37.8	1.8	5.6	1.8
	5	10	73.5	2.9	48.1	2.4	1.1	2.4	73.5	3.0	48.1	2.4	1.0	0.8	76.5	2.9	50.5	2.4	3.5	2.4
40	5	10	98.0	3.0	48.1	2.4	3.3	2.4	98.0	3.0	48.1	2.4	3.2	1.1	101.0	3.0	50.5	2.4	6.3	2.4
	7	10	137.3	4.1	72.8	3.5	2.8	3.5	137.3	4.2	72.8	3.6	2.7	1.2	141.5	4.1	76.4	3.5	6.7	3.5
55	5	10	134.3	3.0	48.1	2.4	3.7	2.4	134.3	3.1	48.1	2.4	3.6	1.4	137.3	3.0	50.5	2.4	6.3	2.4
	7	10	188.0	4.2	72.8	3.5	4.7	3.5	188.0	4.3	72.8	3.6	4.6	1.7	192.2	4.2	76.4	3.5	8.6	3.5
	10	10	256.8	5.9	82.4	4.1	4.1	4.1	256.8	6.1	82.4	4.2	4.0	1.6	262.8	5.9	86.6	4.1	8.7	4.1
60	5	10	146.8	3.0	48.1	2.4	3.8	2.4	146.8	3.1	48.1	2.4	3.6	1.5	149.8	3.0	50.5	2.4	5.9	2.4
	7	10	205.3	4.2	101.8	3.6	15.2	3.6	205.3	4.3	101.8	3.6	15.2	2.0	209.5	4.2	105.4	3.6	20.2	3.6
	13	30	88.6	3.8	33.0	1.5	0.2	1.5	88.6	4.0	33.0	1.6	0.2	0.8	104.4	3.8	34.5	1.5	1.7	1.5
80	10	10	378.2	5.9	377.7	5.9	42.4	5.9	378.2	6.1	377.7	6.1	42.5	2.8	384.2	5.9	383.7	5.9	62.7	5.9
90	13	10	570.6	7.7	570.6	7.7	112.2	7.7	570.6	7.9	570.6	7.9	112.2	3.5	578.4	7.7	578.4	7.7	160.2	7.7
100	15	10	731.6	8.9	731.6	8.9	132.9	8.9	731.6	9.1	731.6	9.1	132.9	4.0	740.6	8.9	740.6	8.9	201.8	8.9
120	15	10	513.2	8.9	513.2	8.9	122.5	8.9	513.2	9.1	513.2	9.1	122.6	4.1	522.2	8.9	522.2	8.9	196.5	8.9
150	15	10	614.2	8.9	614.2	8.9	210.3	8.9	614.2	9.2	614.2	9.2	210.5	4.8	623.2	9.0	623.2	9.0	298.2	9.0
Sum/Avg.		170	256.3	4.9	203.8	4.2	39.0	4.2	256.3	5.1	203.8	4.3	39.0	2.0	263.4	4.9	208.0	4.2	58.6	4.2

Tables 3 and 4 depict and compare in more detail the contribution achieved by grouping identical vessels and berths as well as by fixing variables, respectively. In these tables, columns *opt*, *t(s)*, and *nd* report, per group of instances, the number of problem instances solved to proven optimality, the average execution time in seconds, and the average number of explored nodes, respectively. In addition, columns *cols* and *rows* (only for Table 3) under *red(%)* detail the reduction achieved by applying such improvements.

In Table 3, we report the results for those instances where identical vessels and berths were identified, i.e, instances with up to 80 vessels. Concerning the performance of the studied methods, all of them are able to solve to optimality all 130 instances with up to 80 vessels within the time limit of 2 hours. The results also indicate that grouping identical vessels and berths leads to a significant reduction of more than 60% in the number of rows and columns. Further, in most cases, the improved models require fewer nodes to be explored, which entails a computational effort reduction. It is relevant to mention that, on average, the computational times of the models without improvements are halved when the improvements are incorporated. On the other hand, the new formulations, TI and AF, exhibit a slightly better performance than GSPP in terms of *t(s)*.

After reporting the benefits of reducing the models by grouping identical vessels and berths, in Table 4 we compare the performance of the reduced-cost variable-fixing procedure on the above studied enhanced models for all problem instances. This table shows that by applying this technique we can avoid creating, on average, more than 70% of the initial variables, thus we are able to substantially reduce the formulations sizes. Despite this huge reduction in the number of variables, the reduction in the execution times is more moderate. Taking the AF results as an example, it can be seen that a variable reduction of 70% has been achieved while the execution times have been reduced by 15% on average. This can be explained by the fact that there is a time overhead for solving the linear relaxation and identifying and fixing the variables as well as the fact that the remaining subsequent mathematical model is still difficult to solve.

It is also shown in Table 4 that for instances with 120 and 150 vessels the application of the reduced-cost-based variable-fixing method allows solving more problem instances to optimality. In this regard, this table indicates that the AF with fixed variables performs better than the other formulations being able to find optimal solutions for 166 out of 170 instances within less computational time. These results are detailed in Tables 5 and 6.

Tables 5 and 6 detail the results obtained for the large-size instances considering 120 and 150 vessels, respectively. For each mathematical formulation and instance, we report the final lower and upper bounds, *lb* and *ub*, respectively, the percentage gap *gap(%)*, and the computational time *t(s)*. Note that being all input numbers integer, in these tables *lb* could be replaced by $\lceil lb \rceil$, but we opted to keep *lb* to better highlight the differences among the models.

From Table 5, it can be observed that the use of the variable-fixing approach with the GSPP model allows the solver to accelerate and find all the optimal solutions in comparison to the case where this technique is not applied (*i.e.*, instance 120x15-05). Concerning the resolution times, by analyzing instance by instance we can observe that the time required by the variable-fixing method seems to be worth in several cases. These results are even more promising when tackling problem instances considering a traffic of 150 vessels. As can be seen in Table 6, the additional time required to use the variable-fixing method is worth-while in most of the cases. Furthermore, in Table 6 most of the optimal solutions and best upper bounds were reported by using this approach.

Table 7 summarizes the previous results in this work while indicating the best performing model for each instance set. Thus, the table reports, for all nine methods and for each group of benchmark instances from the literature, the number of proven optimally solved instances, *opt*, and the average execution time, *t(s)*. Column *gap(%)* reports the average percentage gap with respect to the best lower bound obtained per each instance. For the benchmark instances of Cordeau et al. (2005) and Lalla-Ruiz et al. (2012), all instances are solved to optimality by all nine methods. Therefore, it can be seen that our reduced-cost-

Table 3: Comparison of formulations with and without grouping vessels and berths

n	m	#inst	Generalized set partitioning								Time-indexed								Arc-flow							
			GSPP			GSPP_+			red(%)		TI			TI_+			red(%)		AF			AF_+			red(%)	
			opt	t(s)	nd	opt	t(s)	nd	cols	rows	opt	t(s)	nd	opt	t(s)	nd	cols	rows	opt	t(s)	nd	opt	t(s)	nd	cols	rows
30	3	10	10	12.1	84.7	10	9.8	43.0	18.2	0.2	10	13.0	53.5	10	9.8	48.6	18.2	0.3	10	19.0	247.3	10	15.3	353.2	17.5	0.3
	5	10	10	14.4	0.0	10	10.1	1.3	34.5	19.9	10	10.7	0.7	10	9.4	0.9	34.5	20.0	10	4.8	1.1	10	2.9	0.0	33.9	20.0
40	5	10	10	43.8	36.7	10	16.5	5.6	50.9	20.2	10	49.8	40.8	10	16.0	5.0	50.9	20.2	10	63.2	293.7	10	13.7	330.2	50.0	20.2
	7	10	10	38.2	0.0	10	22.6	0.0	47.0	14.5	10	26.4	0.8	10	14.1	0.0	47.0	14.5	10	9.2	1.0	10	4.7	0.0	46.0	14.5
55	5	10	10	64.8	22.7	10	15.3	6.0	64.2	20.6	10	69.1	61.4	10	12.5	3.7	64.2	20.6	10	25.1	12.9	10	11.0	145.1	63.2	20.6
	7	10	10	103.8	167.0	10	29.1	18.9	61.3	14.8	10	106.9	207.4	10	18.1	20.2	61.3	14.8	10	56.5	145.4	10	14.0	135.1	60.3	14.8
	10	10	10	110.6	0.0	10	22.4	2.9	67.9	30.2	10	48.5	8.6	10	10.0	0.0	67.9	30.2	10	31.2	28.9	10	6.2	0.0	67.0	30.2
60	5	10	10	100.0	69.3	10	21.6	150.7	67.2	20.8	10	81.4	107.3	10	17.0	166.4	67.2	20.8	10	68.2	295.8	10	16.0	344.9	66.3	20.8
	7	10	10	220.7	440.8	10	71.1	286.3	50.4	14.7	10	240.5	504.3	10	91.7	663.6	50.4	14.7	10	175.0	214.9	10	79.4	458.0	49.7	14.7
	13	30	30	12.5	0.0	30	1.9	0.0	62.8	60.6	30	3.3	0.0	30	0.9	0.0	62.8	60.6	30	4.1	0.0	30	0.9	0.0	66.9	60.6
80	10	10	10	157.6	19.5	10	165.4	16.6	0.1	0.0	10	113.6	54.6	10	135.8	46.3	0.1	0.0	10	141.6	260.4	10	110.1	251.8	0.1	0.0
Sum/Avg.		130	130	69.5	64.7	130	30.0	40.9	46.3	26.2	130	59.2	80.0	130	25.9	73.4	46.3	26.4	130	46.6	115.5	130	21.3	155.3	46.8	26.2

Table 4: Comparison of formulations with and without variable-fixing

n	m	#inst	Generalized set partitioning							Time-indexed							Arc-flow						
			GSPP_+			$\text{GSPP}_+^{\text{rc}}$			red(%)	TI_+			TI_+^{rc}			red(%)	AF_+			AF_+^{rc}			red(%)
			opt	t(s)	nd	opt	t(s)	nd	cols	opt	t(s)	nd	opt	t(s)	nd	cols	opt	t(s)	nd	opt	t(s)	nd	cols
30	3	10	10	9.8	43.0	10	7.0	32.3	90.9	10	9.8	48.6	10	2.5	13.5	90.9	10	15.3	353.2	10	4.0	285.5	85.3
	5	10	10	10.1	1.3	10	10.2	0.0	97.7	10	9.4	0.9	10	1.1	0.0	97.8	10	2.9	0.0	10	0.7	0.0	93.1
40	5	10	10	16.5	5.6	10	10.6	9.5	93.1	10	16.0	5.0	10	3.1	4.7	93.3	10	13.7	330.2	10	1.4	15.0	87.6
	7	10	10	22.6	0.0	10	20.6	1.3	96.2	10	14.1	0.0	10	1.8	0.0	96.2	10	4.7	0.0	10	1.4	0.0	91.2
55	5	10	10	15.3	6.0	10	10.8	3.8	92.3	10	12.5	3.7	10	3.5	1.1	92.5	10	11.0	145.1	10	1.0	3.1	87.5
	7	10	10	29.1	18.9	10	23.2	35.4	93.5	10	18.1	20.2	10	5.2	14.0	93.7	10	14.0	135.1	10	3.0	168.8	88.8
	10	10	10	22.4	2.9	10	26.2	0.0	95.0	10	10.0	0.0	10	2.6	99.0	95.1	10	6.2	0.0	10	1.8	1.1	89.9
60	5	10	10	21.6	150.7	10	15.1	201.8	92.2	10	17.0	166.4	10	5.9	47.2	92.6	10	16.0	344.9	10	1.6	93.1	88.3
	7	10	10	71.1	286.3	10	70.5	498.4	85.1	10	91.7	663.6	10	44.9	499.9	85.1	10	79.4	458.0	10	25.5	504.4	80.8
	13	10	30	1.9	0.0	30	2.4	0.0	99.4	30	0.9	0.0	30	0.2	0.0	99.4	30	0.9	0.0	30	0.4	0.0	95.0
80	10	10	10	165.4	16.6	10	197.0	17.9	88.8	10	135.8	46.3	10	30.8	49.1	88.7	10	110.1	251.8	10	22.1	194.1	83.7
90	13	10	10	643.7	1073.0	10	927.5	739.9	80.3	10	327.8	1741.7	10	200.2	1647.0	80.3	10	488.8	2754.2	10	215.2	1739.7	72.3
100	15	10	10	1194.0	2824.8	10	1772.2	3837.5	81.8	10	797.2	5255.4	10	541.5	4917.9	81.8	10	856.1	5924.9	10	474.4	3906.8	72.7
120	15	10	9	1791.7	1420.9	10	1744.9	892.6	76.1	10	1063.8	552.0	10	1406.6	1288.2	76.1	10	1231.2	864.9	10	919.3	1523.9	62.4
150	15	10	5	5637.7	9627.5	5	5521.1	4606.0	65.8	3	6067.5	3089.1	3	5878.5	7915.2	65.7	4	5646.1	15325.1	6	5546.7	7381.8	52.5
Sum/Avg.		170	164	568.0	910.4	165	609.6	639.8	80.9	163	505.5	681.9	163	478.2	970.4	80.9	164	499.9	1581.6	166	424.7	930.4	71.9

Table 5: Results for large-size instances considering 120 vessels. In **boldface**, best upper bound value for each instance

n m id	Generalized set partitioning								Time-indexed								Arc-flow							
	GSPP$_+$				GSPP$^{rc}_+$				TI$_+$				TI$^{rc}_+$				AF$_+$				AF$^{rc}_+$			
	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)
120 15 01	4065.0	**4065**	0.00	924.9	4065.0	**4065**	0.00	1133.3	4065.0	**4065**	0.00	1185.4	4065.0	**4065**	0.00	271.1	4065.0	**4065**	0.00	756.5	4065.0	**4065**	0.00	140.2
02	3653.0	**3653**	0.00	550.5	3653.0	**3653**	0.00	847.1	3653.0	**3653**	0.00	123.5	3653.0	**3653**	0.00	70.3	3653.0	**3653**	0.00	47.6	3653.0	**3653**	0.00	18.3
03	3756.0	**3756**	0.00	774.0	3756.0	**3756**	0.00	995.7	3756.0	**3756**	0.00	527.7	3756.0	**3756**	0.00	1128.0	3756.0	**3756**	0.00	98.2	3756.0	**3756**	0.00	243.2
04	3211.0	**3211**	0.00	1390.5	3211.0	**3211**	0.00	1700.4	3211.0	**3211**	0.00	1652.4	3211.0	**3211**	0.00	1421.7	3211.0	**3211**	0.00	1003.6	3211.0	**3211**	0.00	2324.8
05	4294.5	4298	0.08	tlim	4296.0	**4296**	0.00	4196.9	4296.0	**4296**	0.00	2303.1	4296.0	**4296**	0.00	4710.3	4296.0	**4296**	0.00	3727.2	4296.0	**4296**	0.00	3021.3
06	4512.0	**4512**	0.00	758.2	4512.0	**4512**	0.00	1965.8	4512.0	**4512**	0.00	1853.1	4512.0	**4512**	0.00	2162.2	4512.0	**4512**	0.00	1680.9	4512.0	**4512**	0.00	294.1
07	3463.0	**3463**	0.00	596.0	3463.0	**3463**	0.00	768.2	3463.0	**3463**	0.00	126.2	3463.0	**3463**	0.00	96.7	3463.0	**3463**	0.00	36.7	3463.0	**3463**	0.00	49.7
08	3872.0	**3872**	0.00	1202.8	3872.0	**3872**	0.00	1062.5	3872.0	**3872**	0.00	916.4	3872.0	**3872**	0.00	262.6	3872.0	**3872**	0.00	1437.2	3872.0	**3872**	0.00	137.7
09	4176.0	**4176**	0.00	1582.2	4176.0	**4176**	0.00	2043.7	4176.0	**4176**	0.00	1072.5	4176.0	**4176**	0.00	1103.4	4176.0	**4176**	0.00	1974.5	4176.0	**4176**	0.00	1098.9
10	3880.0	**3880**	0.00	2938.3	3880.0	**3880**	0.00	2735.1	3880.0	**3880**	0.00	877.8	3880.0	**3880**	0.00	2839.9	3880.0	**3880**	0.00	1550.1	3880.0	**3880**	0.00	1864.5
Avg.	3888.3	3888.6	0.01	1791.7	3888.4	3888.4	0.00	1744.9	3888.4	3888.4	0.00	1063.8	3888.4	3888.4	0.00	1406.6	3888.4	3888.4	0.00	1231.2	3888.4	3888.4	0.00	919.3

Table 6: Results for large-size instances considering 150 vessels. In **boldface**, best upper bound value for each instance

n m id	Generalized set partitioning								Time-indexed								Arc-flow							
	GSPP$_+$				GSPP$^{\mathrm{rc}}_+$				TI$_+$				TI$^{\mathrm{rc}}_+$				AF$_+$				AF$^{\mathrm{rc}}_+$			
	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)
150 15 01	8214.5	8225	0.13	tlim	8214.6	8220	0.07	tlim	8216.2	8220	0.05	tlim	8214.5	8221	0.08	tlim	8216.9	8220	0.04	tlim	8219.0	**8219**	0.00	7003.8
02	6736.8	6752	0.22	tlim	6737.2	6748	0.16	tlim	6736.9	6744	0.11	tlim	6736.6	6745	0.13	tlim	6737.4	6746	0.13	tlim	6737.7	**6742**	0.06	tlim
03	4655.0	**4655**	0.00	922.4	4655.0	**4655**	0.00	1151.8	4655.0	**4655**	0.00	284.5	4655.0	**4655**	0.00	188.3	4655.0	**4655**	0.00	51.0	4655.0	**4655**	0.00	43.2
04	7303.0	**7303**	0.00	7197.0	7303.0	**7303**	0.00	5172.1	7301.2	7305	0.05	tlim	7301.1	7341	0.54	tlim	7301.5	7309	0.10	tlim	7301.5	7305	0.05	tlim
05	6563.0	**6563**	0.00	2665.3	6561.6	**6563**	0.02	tlim	6561.6	**6563**	0.02	tlim	6561.5	**6563**	0.02	tlim	6563.0	**6563**	0.00	3761.3	6563	**6563**	0.00	2234.0
06	6347.6	**6359**	0.18	tlim	6347.5	6360	0.20	tlim	6347.3	6375	0.43	tlim	6347.4	6361	0.21	tlim	6347.6	6363	0.24	tlim	6348.0	6360	0.19	tlim
07	6343.0	**6343**	0.00	5503.4	6343.0	**6343**	0.00	5432.6	6343.0	**6343**	0.00	5174.9	6343.0	**6343**	0.00	5689.4	6343.0	**6343**	0.00	4914.7	6343.0	**6343**	0.00	6983.2
08	7939.0	**7940**	0.01	tlim	7940.0	**7940**	0.00	5340.6	7939.0	**7940**	0.01	tlim	7939.0	**7940**	0.01	tlim	7939.0	**7940**	0.01	tlim	7940.0	**7940**	0.00	7011.9
09	8242.0	**8242**	0.00	4086.0	8242.0	**8242**	0.00	2114.3	8242.0	**8242**	0.00	4815.8	8242.0	**8242**	0.00	2507.5	8242.0	**8242**	0.00	4533.8	8242.0	**8242**	0.00	3390.9
10	6012.0	6024	0.20	tlim	6012.9	**6016**	0.05	tlim	6012.1	**6016**	0.07	tlim	6012.0	6020	0.13	tlim	6012.4	6020	0.13	tlim	6012.5	**6016**	0.06	tlim
Avg.	6835.6	6840.6	0.07	5637.7	6835.7	6839.0	0.05	5521.1	6835.4	6840.3	0.07	6067.5	6835.2	6843.1	0.11	5878.5	6835.8	6840.1	0.06	5646.1	6836.2	6838.5	0.04	5546.7

based variable-fixing method on top of TI and AF provides the smallest average execution times. Concerning the large-size instances of Nishi et al. (2016), $\mathrm{AF}_{+}^{\mathrm{rc}}$ solves more instances among all methods (46 out of 50), with the smallest average execution time and with the best quality upper bounds. In general words, our experiments indicate that $\mathrm{AF}_{+}^{\mathrm{rc}}$ performed better than all other methods. It is worth noting that the remaining 4 instances were solved to proven optimality by Nishi et al. (2016) in large computing times. The authors proved optimality for 9 out of 10 instances with 150 vessels by running GSPP formulation for approximately 25000 seconds on average. The only open instance left by Nishi et al. (2016), i.e., instance 150x15-08, has been solved to proven optimality by our enhanced models in less than two hours, so all optimal solutions are now assessed for these benchmarks. That drove us to create the new set of very large instances that is evaluated in the next section.

5.3. Computational results on the new set of instances

With the aim of studying the performance of our approaches in larger scenarios, a new set of problem instances is proposed. This new set was generated following the scheme provided in Cordeau et al. (2005). We considered values of the pair (n, m) in $\{(200, 15); (250, 20)\}$, and for each pair we generated 10 instances, obtaining a total of 20 new instances.

Table 8 presents the results for the instances with 200 vessels and 15 berths, while Table 9 does the same for the instances with 250 vessels and 20 berths. In those tables, columns labeled as *lb*, *ub*, *gap(%)* and *t(s)* indicate, for each instance and method, the final lower and upper bounds, the percentage gap, and the computational time, respectively. Note that we do not report results for GSPP, GSPP_{+}, $\mathrm{GSPP}_{+}^{\mathrm{rc}}$, TI and AF. The methods based on the GSPP formulation were not able to solve any of the new instances due to memory limit that can be inputed to the need of storing matrices A and B. Concerning TI and AF, we decided not to test them as they are outperformed by TI_{+} and AF_{+}, as indicated in the previous experiments reported in Tables 2 and 3.

For the instances with 200 vessels, it can be observed that the methods considering the reduced-cost variable-fixing preprocessing tend to be, on average, faster than their versions without such preprocessing. Among the 10 instances of this group, the approaches based on the TI formulation were able to solve to optimality 3 problem instances, while 4 instances were solved to optimality by the methods based on AF. Slight improvements can be noticed in the lower bound values when the reduced-cost variable-fixing technique is applied.

With regards to the instances with 250 vessels, only 1 out of 10 could be solved to proven optimality, i.e., instance 250x20-06 by AF_{+}. For all other instances, the time limit

Table 7: Summary of all reported results. For each instance set, the best value at each category (i.e., opt, t(s), and gap (%)) is **boldfaced**

Inst. group	#inst	Generalized set partitioning									Time indexed									Arc flow									Best method
		GSPP			$GSPP_+$			$GSPP^{rc}_+$			TI			TI_+			TI^{rc}_+			AF			AF_+			AF^{rc}_+			
		opt	t(s)	gap (%)	opt	t(s)	gap (%)	opt	t(s)	gap (%)	opt	t(s)	gap (%)	opt	t(s)	gap (%)	opt	t(s)	gap (%)	opt	t(s)	gap (%)	opt	t(s)	gap (%)	opt	t(s)	gap (%)	
A[1]	30	**30**	12.5	**0.00**	**30**	1.9	**0.00**	**30**	2.4	**0.00**	**30**	3.3	**0.00**	**30**	0.9	**0.00**	**30**	**0.2**	**0.00**	**30**	4.1	**0.00**	**30**	0.9	**0.00**	**30**	0.4	**0.00**	TI^{rc}_+
B[2]	90	**90**	78.7	**0.00**	**90**	24.3	**0.00**	**90**	21.6	**0.00**	**90**	71.8	**0.00**	**90**	22.0	**0.00**	**90**	7.8	**0.00**	**90**	50.3	**0.00**	**90**	18.2	**0.00**	**90**	**4.5**	**0.00**	AF^{rc}_+
C[3]	50	43	1872.4	0.70	44	1886.5	0.69	45	2032.6	0.41	43	1600.8	0.61	43	1678.4	0.61	43	1611.5	0.97	44	1703.5	0.57	44	1666.5	0.57	**46**	**1435.5**	**0.33**	AF^{rc}_+
Avg.	170	163	594.6	0.70	164	568.0	0.69	165	609.6	0.41	163	509.4	0.61	163	505.5	0.61	163	478.2	0.97	164	528.4	0.57	164	499.9	0.57	**166**	**424.7**	**0.33**	AF^{rc}_+

benchmark instances proposed by: [1]Cordeau et al. (2005); [2]Lalla-Ruiz et al. (2012); [3]Nishi et al. (2016)

of 2 hours was reached without proof of optimality. Analyzing these results, it can be seen that for very large instances AF methods face difficulties in improving the given initial upper bounds, differently from TI ones which managed to improve some of them. In terms of best solutions, $\mathrm{TI}_{+}^{\mathrm{rc}}$ is superior to the other methods because it provides 7 best solutions while TI_{+}, AF_{+} and $\mathrm{AF}_{+}^{\mathrm{rc}}$ provide 5, 4 and 3, respectively. In terms of percentage gaps, all methods were able to obtain low average gaps.

6. Conclusions

In this work, we have addressed the dynamic berth allocation problem (DBAP) from a mathematical modeling perspective by providing and assessing two novel time-indexed (TI) and arc-flow (AF) formulations. We have also proposed modeling enhancements aimed at grouping similar berths and vessels, and a variable-fixing procedure based on reduced costs. Extensive computational experiments on benchmark instances have been performed to evaluate the investigated methods and compare them with the best model from the literature, that is, the generalized set-partitioning problem (GSPP) formulation.

The AF formulation performs better than the TI, which in turn performs better than the GSPP. Therefore, AF is advisable when used as a standalone model. The proposed modeling enhancements based on grouping similar vessels and berths improved the performance of all models by reducing the number of variable and leading to better computational times. The reduced-cost variable-fixing procedure leads to further improvements. In particular, by applying this procedure to the AF model we could solve to proven optimality 166 out of 170 benchmark instances from the literature within two hours of time limit, including the only instance that was still open. Based on these findings, the AF formulation has shown superiority compared to the other formulations, and thus, it is recommended to be used in real-world environments. Taking into account the good results on the benchmark instances, a new set containing 20 large-sized instances involving between 200 and 250 vessels has been proposed. The GSPP models were not able to deal with these new instances due to excessive memory requirements. On the contrary, models based on TI and AF did not show memory problems and managed to optimally solve 5 instances and provide small gaps for the other ones.

As future work, we plan to adapt these novel formulations to other maritime logistics problems where the berth allocation is involved. In this sense, the joint consideration of the quay crane allocation and scheduling problem with berth scheduling, such as the one in Agra and Oliveira (2018), and continuous berth allocation problems, (see e.g. Frojan et al. 2015), appear to be interesting future research directions.

Table 8: Results for new large-size instances considering 200 vessels. In **boldface**, best upper bound value for each instance

			Time-indexed								Arc-flow							
n	*m*	id	TI_+				TI^{rc}_+				AF_+				AF^{rc}_+			
			lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)
200	15	01	12603.3	12613	0.08	tlim	12603.3	**12609**	0.05	tlim	12603.3	12709	0.83	tlim	12603.3	12709	0.83	tlim
		02	10319.0	**10319**	0.00	6301.8	10317.6	**10319**	0.01	tlim	10319.0	**10319**	0.00	3620.8	10319.0	**10319**	0.00	1911.0
		03	11289.8	11558	2.32	tlim	11295.4	**11355**	0.52	tlim	11292.0	11558	2.30	tlim	11294.4	11416	1.07	tlim
		04	15433.9	15480	0.30	tlim	15437.4	**15441**	0.02	tlim	15441.0	**15441**	0.00	6880.6	15441.0	**15441**	0.00	3636.4
		05	18164.7	**18352**	1.02	tlim	18165.0	**18352**	1.02	tlim	18159.4	**18352**	1.05	tlim	18165.6	**18352**	1.02	tlim
		06	16869.0	**16869**	0.00	6836.4	16869.0	**16869**	0.00	6454.6	16869.0	**16869**	0.00	1015.5	16869.0	**16869**	0.00	1575.9
		07	13023.7	**13226**	1.53	tlim	13023.7	**13226**	1.53	tlim	13023.5	**13226**	1.53	tlim	13025.0	**13226**	1.52	tlim
		08	14176.6	14537	2.48	tlim	14181.2	14537	2.45	tlim	14180.4	14298	0.82	tlim	14180.5	**14259**	0.55	tlim
		09	18115.8	18198	0.45	tlim	18118.0	**18118**	0.00	1911.5	18118.0	**18118**	0.00	3879.7	18118.0	**18118**	0.00	4951.6
		10	17095.4	17263	0.97	tlim	17100.9	17263	0.94	tlim	17100.8	**17118**	0.10	tlim	17101.6	17134	0.19	tlim
Avg.			14709.1	14841.5	0.91	7073.8	14711.2	14808.9	0.65	6596.6	14710.6	14800.8	0.66	5859.7	14711.7	14784.3	0.52	5527.5

Table 9: Results for new large-size instances considering 250 vessels. In **boldface**, best upper bound value for each instance

			Time-indexed								Arc-flow							
n	*m*	id	TI_+				TI^{rc}_+				AF_+				AF^{rc}_+			
			lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)	lb	ub	gap (%)	t(s)
250	20	01	15632.2	**15769**	0.87	tlim	15632.4	**15769**	0.87	tlim	15632.3	**15769**	0.87	tlim	15632.3	**15769**	0.87	tlim
		02	15774.9	**15915**	0.88	tlim	15775.1	**15915**	0.88	tlim	15774.9	**15915**	0.88	tlim	15775.0	**15915**	0.88	tlim
		03	16518.9	**16606**	0.52	tlim	16518.8	16631	0.67	tlim	16518.9	16724	1.23	tlim	16518.9	16724	1.23	tlim
		04	16422.6	16490	0.41	tlim	16422.6	**16481**	0.35	tlim	16422.6	16509	0.52	tlim	16422.6	16509	0.52	tlim
		05	15661.0	**15837**	1.11	tlim	15661.0	**15837**	1.11	tlim	15661.0	**15837**	1.11	tlim	15661.0	**15837**	1.11	tlim
		06	20060.0	20193	0.66	tlim	20060.0	20193	0.66	tlim	20060.0	**20060**	0.00	5180.6	20060.0	20193	0.66	tlim
		07	14283.3	14514	1.59	tlim	14283.3	**14362**	0.55	tlim	14283.3	14514	1.59	tlim	14283.3	14514	1.59	tlim
		08	16303.7	16386	0.50	tlim	16304.1	**16383**	0.48	tlim	16303.7	16498	1.18	tlim	16303.7	16498	1.18	tlim
		09	15863.5	16121	1.60	tlim	15863.5	**15917**	0.34	tlim	15863.5	16121	1.60	tlim	15863.5	16121	1.60	tlim
		10	16282.5	**16371**	0.54	tlim	16282.5	16428	0.89	tlim	16282.5	16428	0.89	tlim	16282.5	16428	0.89	tlim
Avg.			16280.3	16420.2	0.87	7200.0	16280.3	16391.6	0.68	7200.0	16280.3	16437.5	0.99	6998.1	16280.3	16450.8	1.05	7200.0

Acknowledgments

This research was partially funded by the CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil, grant No. 234814/2014-4.

References

Agra, A., Oliveira, M., 2018. MIP approaches for the integrated berth allocation and quay crane assignment and scheduling problem. European Journal of Operational Research 264 (1), 138 – 148.

Baldacci, R., Christofides, N., Mingozzi, A., 2008. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming 115 (2), 351–385.

Baptiste, P., Della Croce, F., Grosso, A., Tkindt, V., 2010. Sequencing a single machine with due dates and deadlines: an ILP-based approach to solve very large instances. Journal of Scheduling 13 (1), 39–47.

Bierwirth, C., Meisel, F., 2010. A survey of berth allocation and quay crane scheduling problems in container terminals. European Journal of Operational Research 202 (3), 615–627.

Bierwirth, C., Meisel, F., 2015. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. European Journal of Operational Research 244 (3), 675–689.

Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., Lusby, R., 2011. Models for the discrete berth allocation problem: a computational comparison. Transportation Research Part E 47 (4), 461–473.

Christensen, C., Holst, C., 2008. Berth allocation in container terminals. Master's thesis, Technical University of Denmark.

Cordeau, J.-F., Gaudioso, M., Laporte, G., Moccia, L., 2007. The service allocation problem at the Gioia Tauro maritime terminal. European Journal of Operational Research 176 (2), 1167–1184.

Cordeau, J.-F., Laporte, G., Legato, P., Moccia, L., 2005. Models and tabu search heuristics for the berth-allocation problem. Transportation Science 39 (4), 526–538.

De Oliveira, R. M., Mauri, G. R., Lorena, L. A. N., 2012. Clustering search for the berth allocation problem. Expert Systems with Applications 39 (5), 5499–5505.

Delorme, M., Iori, M., Martello, S., 2016. Bin packing and cutting stock problems: Mathematical models and exact algorithms. European Journal of Operational Research 255 (1), 1–20.

Fischetti, M., Monaci, M., 2014. Exploiting erraticism in search. Operations Research 62 (1), 114–122.

Frojan, P., Correcher, J. F., Alvarez-Valdes, R., Koulouris, G., Tamarit, J. M., 2015. The continuous berth allocation problem in a container terminal with multiple quays. Expert Systems with Applications 42 (21), 7356 – 7366.

Gao, C., Lu, G., Yao, X., Li, J., 2017. An iterative pseudo-gap enumeration approach for the multidimensional multiple-choice knapsack problem. European Journal of Operational Research 260 (1), 1–11.

Ibaraki, T., Nakamura, Y., 1994. A dynamic programming method for single machine scheduling. European Journal of Operational Research 76 (1), 72 – 82.

Imai, A., Nishimura, E., Papadimitriou, S., 2001. The dynamic berth allocation problem for a container port. Transportation Research Part B: Methodological 35 (4), 401–417.

Kramer, A., Dell'Amico, M., Iori, M., 2018. Enhanced arc-flow formulations to minimize weighted completion time on identical parallel machines. CoRR abs/1808.10661, technical report available at: http://arxiv.org/abs/1808.10661.

Lalla-Ruiz, E., 2016. Intelligent management of seaside logistic operations at maritime container terminals. 4OR 2 (15), 217–218.

Lalla-Ruiz, E., Melián-Batista, B., Moreno-Vega, J. M., 2012. Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. Engineering Applications of Artificial Intelligence 25 (6), 1132–1141.

Lalla-Ruiz, E., Melián-Batista, B., Moreno-Vega, J. M., 2016. A cooperative search for berth scheduling. The Knowledge Engineering Review 31 (5), 498–507.

Lalla-Ruiz, E., Voß, S., 2016a. Improving solver performance through redundancy. Journal of Systems Science and Systems Engineering 25 (3), 303–325.

Lalla-Ruiz, E., Voß, S., Feb 2016b. POPMUSIC as a matheuristic for the berth allocation problem. Annals of Mathematics and Artificial Intelligence 76 (1), 173–189.

Lodi, A., Tramontani, A., 2013. Performance variability in mixed-integer programming. In: Topaloglu, H. (Ed.), TutORials in Operations Research: Theory Driven by Influential Applications. INFORMS, pp. 1–12.

Mauri, G. R., Ribeiro, G. M., Lorena, L. A. N., Laporte, G., 2016. An adaptive large neighborhood search for the discrete and continuous berth allocation problem. Computers & Operations Research 70, 140–154.

Nishi, T., Okura, T., Lalla-Ruiz, E., Voß, S., 2016. A dynamic programming-based matheuristic for the dynamic berth allocation problem. Annals of Operations Research, 1–20.

Notteboom, T., 2006. The time factor in liner shipping services. Maritime Economics & Logistics 8 (1), 19–39.

Pessoa, A., Uchoa, E., de Aragão, M. P., Rodrigues, R., Dec 2010. Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. Mathematical Programming Computation 2 (3), 259–290.

Savelsbergh, M. W., 1994. Preprocessing and probing techniques for mixed integer programming problems. ORSA Journal on Computing 6 (4), 445–454.

Sousa, J. P., Wolsey, L. A., 1992. A time indexed formulation of non-preemptive single machine scheduling problems. Mathematical Programming 54 (1), 353–367.

Stahlbock, R., Voß, S., 2008. Vehicle routing problems and container terminal operations– an update of research. In: The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, pp. 551–589.

Steenken, D., Voß, S., Stahlbock, R., 2004. Container terminal operations and operations research – a classification and literature review. OR Spectrum 26 (1), 3–49.

Ting, C.-J., Wu, K.-C., Chou, H., 2014. Particle swarm optimization algorithm for the berth allocation problem. Expert Systems with Applications 41 (4), 1543–1550.

Valério de Carvalho, J., 1999. Exact solution of bin-packing problems using column generation and branch-and-bound. Annals of Operations Research 86 (0), 629–659.

Valério de Carvalho, J., 2002. LP models for bin packing and cutting stock problems. European Journal of Operational Research 141 (2), 253–273.

Appendix A. Detailed results

In this appendix, we report detailed results obtained by all the mathematical programming approaches studied in the paper. Table A.10 details the results for the set of 90 instances proposed by Lalla-Ruiz et al. (2012). The results for the 30 instances proposed by Cordeau et al. (2005) are shown in Table A.11, while Table A.12 reports the results for the 50 benchmark instances of Nishi et al. (2016). Finally, Table A.13 contains the results for the set of 20 new instances proposed in this work. In all tables, we report the best lower (column *lb*) and upper (column *ub*) bounds found for each instance (considering all mathematical programming approaches). For each method and instance columns *t(s)* and *nd* present the total execution time in seconds and the number of explored nodes, respectively. The last line of each table reports average values.

Table A.10: Detailed results for Lalla-Ruiz et al. (2012) benchmark instances

					Generalized set partitioning						Time indexed						Arc-flow					
n	m	id	lb	ub	GSPP		GSPP_+		$\text{GSPP}^{\text{rc}}_+$		TI		TI_+		TI^{rc}_+		AF		AF_+		AF^{rc}_+	
					t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd
30	3	01	1763	1763	10.4	0	7.2	0	5.3	0	10.5	0	5.5	0	1.5	0	20.3	178	12.1	133	0.6	0
		02	2090	2090	17.0	9	11.5	86	8.6	5	17.5	22	13.2	23	3.6	12	11.1	654	15.6	648	4.2	291
		03	2186	2186	7.3	0	6.6	0	5.8	0	7.4	0	9.6	0	1.4	0	7.7	61	3.7	0	0.6	0
		04	1538	1538	8.1	0	7.2	0	5.3	0	4.6	0	5.2	0	1.7	0	6.6	37	4.9	15	0.8	0
		05	2114	2114	15.6	0	14.0	13	7.5	11	23.2	11	10.4	53	3.9	9	10.3	60	5.8	75	1.7	11
		06	2185	2185	12.2	0	15.4	0	6.5	0	14.1	0	15.3	0	2.5	0	95.5	1069	76.5	1486	10.7	702
		07	1845	1845	16.7	50	12.9	46	9.3	53	18.7	65	16.3	130	4.9	36	20.8	221	12.3	205	10.1	1043
		08	1271	1271	7.8	28	5.7	12	5.2	50	7.6	9	6.8	35	1.0	0	4.9	22	4.4	222	0.7	0
		09	1595	1595	14.7	760	10.9	273	9.4	204	15.4	428	12.0	245	2.7	78	10.9	171	15.6	748	10.5	808
		10	2195	2195	11.2	0	6.6	0	7.0	0	11.0	0	3.1	0	1.5	0	2.4	0	2.2	0	0.5	0
	5	01	1149	1149	10.7	0	7.6	0	13.5	0	5.3	0	6.3	0	0.7	0	1.8	0	1.2	0	0.7	0
		02	1475	1475	21.6	0	12.1	0	11.5	0	20.5	0	11.5	0	1.4	0	3.9	0	3.2	0	0.8	0
		03	1542	1542	25.2	0	12.5	13	10.6	0	13.4	7	15.8	9	1.4	0	9.0	0	2.4	0	0.8	0
		04	1075	1075	12.0	0	8.2	0	8.8	0	4.2	0	5.2	0	0.8	0	3.3	0	2.4	0	0.6	0
		05	1463	1463	11.3	0	12.0	0	9.8	0	11.8	0	7.2	0	1.0	0	4.9	0	2.8	0	0.7	0
		06	1580	1580	12.9	0	11.2	0	8.6	0	20.8	0	12.9	0	1.5	0	4.6	0	5.0	0	0.9	0
		07	1276	1276	10.6	0	9.2	0	8.6	0	5.5	0	8.0	0	0.9	0	3.2	0	2.2	0	0.7	0
		08	870	870	12.5	0	7.5	0	10.3	0	5.9	0	7.0	0	0.8	0	5.8	0	3.1	0	0.6	0
		09	1134	1134	13.1	0	12.0	0	10.0	0	9.4	0	10.6	0	1.2	0	6.2	11	3.5	0	0.8	0
		10	1527	1527	14.0	0	9.3	0	10.3	0	9.8	0	9.2	0	1.2	0	5.4	0	3.5	0	0.7	0
40	5	01	2301	2301	34.0	0	11.2	0	9.2	0	29.5	0	6.6	0	2.0	0	6.3	0	2.0	0	0.7	0
		02	2829	2829	43.2	33	15.7	15	9.8	0	57.4	83	18.0	0	3.4	0	18.3	49	5.7	58	1.1	10
		03	2880	2880	63.5	187	24.0	12	12.3	0	99.0	190	18.6	9	4.0	7	91.1	458	51.4	2083	1.6	8
		04	2001	2001	25.7	0	12.0	0	9.0	34	17.5	0	9.0	0	1.4	0	9.5	0	4.0	0	0.9	0
		05	2815	2815	66.0	102	24.1	13	13.7	24	85.4	43	33.4	18	5.0	9	234.7	1033	9.8	112	2.6	41
		06	2934	2934	66.6	27	24.4	9	15.0	29	72.8	19	21.4	16	5.7	24	195.9	1066	45.5	1033	2.4	69
		07	2632	2632	40.0	0	13.7	0	9.1	0	20.6	0	15.5	0	2.0	0	5.7	0	3.5	0	0.8	0
		08	1835	1835	36.3	0	12.7	0	8.3	0	37.3	50	9.7	0	1.6	0	24.7	212	5.8	16	1.1	11
		09	2086	2086	36.0	18	11.3	7	10.4	8	31.2	9	13.1	7	2.6	7	37.7	119	3.5	0	1.7	11
		10	2962	2962	27.0	0	15.8	0	9.3	0	47.3	14	14.3	0	3.4	0	7.8	0	6.4	0	0.9	0
	7	01	1458	1458	37.1	0	20.2	0	24.3	0	22.5	0	15.3	0	1.9	0	10.9	0	4.3	0	1.7	0
		02	1375	1375	25.7	0	15.1	0	20.0	0	8.9	0	8.4	0	0.8	0	3.7	0	2.3	0	1.1	0
		03	2119	2119	55.2	0	28.9	0	24.0	0	34.9	0	21.0	0	3.7	0	14.6	10	5.1	0	2.2	0
		04	1591	1591	37.9	0	26.3	0	19.8	0	22.2	0	10.6	0	1.5	0	8.6	0	3.1	0	1.4	0
		05	1847	1847	40.3	0	23.5	0	19.2	0	24.6	0	14.6	0	1.5	0	8.5	0	5.1	0	1.3	0
		06	2080	2080	46.6	0	25.4	0	18.6	0	25.2	0	18.0	0	1.6	0	7.6	0	5.2	0	1.1	0
		07	1841	1841	37.7	0	25.0	0	17.9	13	28.1	8	15.1	0	1.7	0	10.4	0	7.7	0	1.3	0
		08	2025	2025	36.9	0	24.8	0	22.4	0	28.5	0	16.7	0	1.7	0	9.7	0	7.3	0	1.2	0
		09	1880	1880	27.4	0	15.5	0	19.4	0	24.7	0	9.6	0	1.1	0	9.8	0	5.0	0	1.1	0

Continued on next page

Table A.10: (*Continued*)

					Generalized set partitioning						Time indexed						Arc-flow					
n	m	id	lb	ub	GSPP		GSPP_+		$\text{GSPP}^{\text{rc}}_+$		TI		TI_+		TI^{rc}_+		AF		AF_+		AF^{rc}_+	
					t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd
		10	1883	1883	37.0	0	20.9	0	20.7	0	44.2	0	11.6	0	2.1	0	8.3	0	2.3	0	1.3	0
55	5	01	4689	4689	86.2	80	17.9	0	13.1	0	156.0	420	14.3	0	4.1	0	45.4	25	5.8	0	1.3	0
		02	5467	5467	52.5	0	9.8	0	9.7	0	58.9	0	4.5	0	2.8	0	10.6	0	3.3	0	0.7	0
		03	5499	5499	57.9	0	24.1	0	11.6	18	73.9	0	17.0	4	3.3	0	8.6	0	6.6	20	0.8	0
		04	4165	4165	67.9	0	14.7	0	10.5	0	48.8	0	9.9	0	3.4	0	51.6	39	11.3	394	1.1	0
		05	5478	5478	55.8	0	16.7	0	10.5	0	46.7	0	14.3	0	3.3	0	21.4	13	3.8	0	0.8	0
		06	5595	5595	63.7	0	17.5	0	10.3	0	56.9	0	4.2	0	3.5	0	10.0	0	3.4	0	0.8	0
		07	4870	4870	27.1	0	8.6	0	10.1	0	17.5	0	4.5	0	2.7	0	13.9	5	2.6	0	0.9	0
		08	3552	3552	145.9	147	27.7	60	12.1	20	143.5	194	29.4	33	5.7	11	57.2	33	59.4	971	2.0	31
		09	4273	4273	54.6	0	8.8	0	10.4	0	33.6	0	12.3	0	3.1	0	8.8	0	10.1	66	1.1	0
		10	5739	5739	36.1	0	7.5	0	9.9	0	55.0	0	14.4	0	2.7	0	23.9	14	4.2	0	0.7	0
	7	01	2846	2846	97.9	0	32.2	0	21.9	0	83.9	70	11.0	0	2.7	0	25.0	29	10.6	13	1.5	0
		02	2883	2883	124.4	109	25.0	8	24.5	0	108.4	13	15.2	0	9.7	0	30.4	29	4.8	0	4.0	10
		03	3825	3825	60.8	0	31.8	0	23.3	0	47.2	0	7.6	0	3.5	0	15.5	0	5.8	0	1.1	0
		04	2951	2951	60.6	0	19.5	0	20.9	0	35.7	0	6.8	0	2.2	0	11.4	0	5.0	0	1.3	0
		05	3797	3797	94.8	99	24.6	25	26.2	20	54.1	20	26.9	9	4.8	0	52.8	49	12.3	72	2.1	0
		06	3783	3783	79.6	0	36.1	0	21.0	0	53.3	0	19.0	0	3.7	8	24.9	22	9.0	20	1.6	12
		07	3774	3774	62.5	0	18.9	0	19.0	0	44.9	0	15.3	0	3.5	0	11.7	0	7.5	0	1.4	0
		08	3862	3862	119.7	60	28.2	0	23.1	0	89.4	16	16.8	0	4.5	0	37.0	50	7.8	0	1.4	0
		09	3591	3591	129.3	998	38.2	0	20.4	0	95.2	66	31.6	12	5.9	49	42.8	77	14.0	122	10.9	1587
		10	3623	3623	208.6	404	36.7	156	31.8	334	456.4	1889	30.6	181	11.3	83	313.4	1198	63.1	1124	4.7	79
	10	01	2742	2742	81.3	0	18.3	0	27.5	0	44.4	0	6.2	0	2.7	0	15.9	0	4.2	0	1.9	0
		02	2527	2527	144.9	0	23.7	0	24.4	0	56.4	0	13.0	0	3.4	0	67.9	86	8.3	0	2.7	11
		03	2544	2544	98.7	0	19.7	0	25.9	0	23.8	0	10.1	0	1.7	0	15.8	0	4.0	0	1.8	0
		04	3315	3315	135.0	0	25.9	0	25.4	0	48.9	0	9.6	0	2.2	0	33.2	4	5.7	0	1.5	0
		05	3109	3109	162.5	0	28.3	0	27.1	0	58.2	0	13.7	0	3.8	0	25.7	0	8.6	0	2.2	0
		06	2283	2283	79.0	0	18.8	0	25.5	0	16.0	0	4.1	0	0.9	0	14.1	0	2.8	0	1.2	0
		07	2144	2144	89.6	0	19.3	0	26.8	0	45.9	0	8.8	0	2.8	0	39.7	107	5.8	0	2.4	0
		08	2720	2720	95.6	0	26.0	29	25.8	0	85.3	86	16.9	0	5.7	990	66.4	92	9.8	0	1.9	0
		09	2149	2149	73.0	0	18.4	0	26.4	0	33.8	0	4.4	0	1.3	0	15.5	0	5.2	0	1.4	0
		10	2814	2814	146.4	0	25.5	0	27.4	0	72.5	0	13.0	0	1.7	0	18.1	0	8.2	0	1.4	0
60	5	01	5753	5753	45.6	0	21.3	0	10.3	0	62.4	0	9.7	0	4.2	0	42.5	41	3.8	0	0.7	0
		02	6884	6884	308.7	573	55.3	1470	56.7	2008	284.0	1049	55.2	1651	23.3	472	392.5	2762	80.9	985	8.0	918
		03	6780	6780	65.3	0	17.8	0	10.3	0	39.8	0	4.5	0	2.6	0	23.4	27	3.8	0	0.6	0
		04	5092	5092	110.3	19	14.3	26	9.4	0	54.4	0	10.3	0	3.6	0	53.0	47	3.9	0	1.1	0
		05	6715	6715	83.5	0	19.7	0	12.0	0	78.2	0	15.4	0	4.7	0	27.5	10	3.6	0	0.8	0
		06	6616	6616	75.3	0	7.1	0	10.0	0	37.5	0	8.5	0	4.6	0	44.0	19	3.1	0	0.6	0
		07	6011	6011	59.6	0	20.2	0	9.8	0	36.0	0	4.7	0	3.8	0	23.5	6	4.8	0	0.7	0
		08	4385	4385	128.6	101	26.0	11	10.9	10	77.6	24	30.6	13	3.8	0	25.5	21	44.9	2438	1.8	13

Continued on next page

Table A.10: (*Continued*)

					Generalized set partitioning						Time indexed						Arc-flow					
n	m	id	lb	ub	GSPP		GSPP_+		$\text{GSPP}_+^{\text{rc}}$		TI		TI_+		TI_+^{rc}		AF		AF_+		AF_+^{rc}	
					t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd
		09	5235	5235	34.2	0	17.5	0	10.3	0	54.2	0	19.9	0	3.8	0	24.9	17	7.4	26	1.1	0
		10	7255	7255	88.9	0	16.7	0	10.9	0	89.8	0	11.2	0	5.0	0	25.7	8	3.9	0	0.6	0
	7	01	3707	3707	118.8	5	88.1	1333	59.7	1271	132.0	60	105.0	1772	26.9	0	34.4	49	22.9	86	6.3	28
		02	4146	4146	426.4	1290	101.6	555	96.1	1222	505.3	2032	104.2	1387	98.1	1893	126.8	223	160.1	1643	18.0	331
		03	4273	4273	301.2	783	78.8	183	58.9	133	273.3	363	83.8	503	32.1	220	111.1	156	53.6	367	30.2	988
		04	3910	3910	120.7	0	50.8	0	38.0	0	106.2	92	37.0	0	13.6	0	76.7	37	40.0	206	3.1	0
		05	4251	4251	238.1	1088	70.0	60	39.9	76	120.0	136	77.7	127	14.9	106	69.6	71	59.7	289	4.1	35
		06	5727	5727	155.8	0	56.0	0	39.5	0	149.8	60	48.0	78	10.3	0	58.1	59	17.9	28	1.4	0
		07	3719	3719	97.1	0	36.3	0	45.6	0	121.7	5	38.7	0	13.1	0	61.9	23	12.8	8	3.0	0
		08	4582	4582	409.0	1062	125.5	635	193.0	1848	566.5	1892	287.7	2611	201.5	2610	983.4	1390	218.6	965	92.0	1283
		09	3979	3979	123.8	11	46.1	4	77.7	320	264.7	200	79.6	75	16.6	39	143.2	101	116.6	466	40.4	1413
		10	4107	4107	216.6	169	58.0	93	56.2	114	165.3	203	55.1	83	22.3	131	84.5	40	91.8	522	56.5	966
Avg.			-	-	78.7	91	24.3	57	21.6	87	71.8	109	22.0	101	7.8	75	50.3	138	18.2	196	4.5	119

Table A.11: Detailed results for Cordeau et al. (2005) benchmark instances

					Generalized set partitioning						Time indexed						Arc-flow					
n	m	id	lb	ub	GSPP		GSPP_+		$\text{GSPP}_+^{\text{rc}}$		TI		TI_+		TI_+^{rc}		AF		AF_+		AF_+^{rc}	
					t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd
60	13	01	1409	1409	13.7	0	1.9	0	2.3	0	3.7	0	2.9	0	0.2	0	6.4	0	1.1	0	0.3	0
		02	1261	1261	11.6	0	1.9	0	2.3	0	2.9	0	0.7	0	0.2	0	4.6	0	1.0	0	0.4	0
		03	1129	1129	11.4	0	1.7	0	2.2	0	2.6	0	0.6	0	0.2	0	3.5	0	1.0	0	0.4	0
		04	1302	1302	12.0	0	1.8	0	2.3	0	3.2	0	0.7	0	0.2	0	2.4	0	0.9	0	0.4	0
		05	1207	1207	12.2	0	1.9	0	2.5	0	2.8	0	0.6	0	0.2	0	3.5	0	0.6	0	0.4	0
		06	1261	1261	13.0	0	1.8	0	2.2	0	2.9	0	1.9	0	0.2	0	2.4	0	1.5	0	0.4	0
		07	1279	1279	11.8	0	2.0	0	2.2	0	2.9	0	0.7	0	0.2	0	2.4	0	1.0	0	0.4	0
		08	1299	1299	12.5	0	1.9	0	2.4	0	3.2	0	0.7	0	0.2	0	3.9	0	0.6	0	0.4	0
		09	1444	1444	12.4	0	2.0	0	2.4	0	3.2	0	0.8	0	0.2	0	3.8	0	0.6	0	0.5	0
		10	1213	1213	12.3	0	1.8	0	2.4	0	3.3	0	0.7	0	0.2	0	2.6	0	1.3	0	0.4	0
		11	1368	1368	13.7	0	2.0	0	2.5	0	3.4	0	0.8	0	0.2	0	5.3	0	1.6	0	0.4	0
		12	1325	1325	13.2	0	1.9	0	2.3	0	3.1	0	0.8	0	0.2	0	5.8	0	1.0	0	0.4	0
		13	1360	1360	12.1	0	1.9	0	2.4	0	3.2	0	0.8	0	0.2	0	4.5	0	1.3	0	0.4	0
		14	1233	1233	13.0	0	1.9	0	2.5	0	3.8	0	0.7	0	0.2	0	5.7	0	1.1	0	0.4	0
		15	1295	1295	12.7	0	1.9	0	2.4	0	3.1	0	1.6	0	0.2	0	5.1	0	0.6	0	0.4	0
		16	1364	1364	12.2	0	1.9	0	2.4	0	4.0	0	0.7	0	0.2	0	4.3	0	1.1	0	0.4	0
		17	1283	1283	12.6	0	1.8	0	2.3	0	2.9	0	1.3	0	0.2	0	4.8	0	0.6	0	0.4	0
		18	1345	1345	12.1	0	2.0	0	2.3	0	3.1	0	1.2	0	0.2	0	5.1	0	0.6	0	0.4	0
		19	1367	1367	11.9	0	2.0	0	2.3	0	3.5	0	0.8	0	0.2	0	3.5	0	1.1	0	0.4	0
		20	1328	1328	13.8	0	2.2	0	2.5	0	5.7	0	1.1	0	0.2	0	6.0	0	0.6	0	0.5	0
		21	1341	1341	12.4	0	1.9	0	2.3	0	3.2	0	0.8	0	0.2	0	4.6	0	0.8	0	0.4	0
		22	1326	1326	12.7	0	2.2	0	2.4	0	3.3	0	1.3	0	0.2	0	3.7	0	0.9	0	0.5	0
		23	1266	1266	11.3	0	1.8	0	2.2	0	2.9	0	0.7	0	0.2	0	4.6	0	0.6	0	0.4	0
		24	1260	1260	12.0	0	1.8	0	2.4	0	2.8	0	0.7	0	0.2	0	4.0	0	0.8	0	0.4	0
		25	1376	1376	13.5	0	2.1	0	2.7	0	3.5	0	0.8	0	0.2	0	2.8	0	0.7	0	0.5	0
		26	1318	1318	11.8	0	1.8	0	2.4	0	4.0	0	0.7	0	0.2	0	4.2	0	1.1	0	0.4	0
		27	1261	1261	11.8	0	1.8	0	2.4	0	2.9	0	0.7	0	0.2	0	3.2	0	1.1	0	0.4	0
		28	1359	1359	12.4	0	2.2	0	2.4	0	3.3	0	0.7	0	0.2	0	2.6	0	1.1	0	0.4	0
		29	1280	1280	12.9	0	1.8	0	2.4	0	3.1	0	0.7	0	0.2	0	3.8	0	0.9	0	0.4	0
		30	1344	1344	13.0	0	1.9	0	2.4	0	3.2	0	0.8	0	0.2	0	5.4	0	1.0	0	0.4	0
Avg.			-	-	12.5	0	1.9	0	2.4	0	3.3	0	0.9	0	0.2	0	4.1	0	0.9	0	0.4	0

Table A.12: Detailed results for Nishi et al. (2016) benchmark instances

					Generalized set partitioning						Time indexed						Arc-flow					
n	m	id	lb	ub	GSPP		GSPP$_+$		GSPP$_+^{\text{rc}}$		TI		TI$_+$		TI$_+^{\text{rc}}$		AF		AF$_+$		AF$_+^{\text{rc}}$	
					t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd
80	10	01	3417	3417	177.4	119	168.1	90	192.2	129	152.0	390	142.6	307	24.9	80	198.3	384	119.8	298	15.2	190
		02	2602	2602	138.5	0	147.3	0	190.5	2	106.7	4	127.6	4	22.8	14	114.6	136	93.1	136	18.0	207
		03	2602	2602	138.5	0	147.0	0	189.8	6	106.3	4	130.2	4	21.0	30	114.6	136	93.2	136	15.9	123
		04	3504	3504	179.2	13	191.0	13	202.0	22	149.8	27	188.3	27	53.1	284	144.0	642	118.0	642	39.8	531
		05	4010	4010	175.5	55	186.6	55	209.8	20	166.8	94	204.9	94	58.5	80	478.7	606	380.0	606	48.2	542
		06	3595	3595	180.9	8	189.7	8	203.3	0	145.8	0	187.0	0	26.5	0	95.7	63	77.7	63	22.0	254
		07	3034	3034	159.0	0	169.8	0	195.8	0	97.0	27	120.9	27	39.2	3	80.2	57	65.0	57	21.0	88
		08	2588	2588	141.4	0	149.4	0	185.0	0	54.9	0	66.8	0	10.2	0	80.9	510	66.5	510	12.3	6
		09	4367	4367	135.4	0	146.9	0	203.0	0	75.6	0	91.7	0	17.7	0	27.5	0	22.5	0	10.5	0
		10	3407	3407	149.7	0	158.3	0	198.4	0	80.7	0	97.8	0	34.6	0	81.7	70	65.6	70	18.2	0
90	13	01	2252	2252	634.3	1754	698.6	1754	946.4	1933	342.5	3961	419.4	3961	167.9	1354	299.9	1709	247.2	1709	290.2	2134
		02	2533	2533	743.4	3705	816.9	3705	1121.5	2237	557.1	5590	662.0	5590	551.1	6441	1520.4	10429	1321.5	10429	313.5	2865
		03	2533	2533	801.7	3705	811.8	3705	1030.5	1991	545.1	5590	627.8	5590	565.9	5511	1330.5	10429	1322.3	10429	659.9	7412
		04	2402	2402	401.0	3	557.2	3	851.4	8	76.2	7	88.8	7	29.6	64	71.1	48	72.2	48	27.7	70
		05	2468	2468	510.6	50	570.6	50	878.7	60	128.9	83	150.9	83	39.5	31	79.8	102	80.9	102	52.1	110
		06	3267	3267	776.0	612	741.2	612	965.6	524	354.4	752	437.0	752	280.5	1244	651.4	1934	659.1	1934	392.4	2431
		07	2115	2115	506.8	7	537.2	7	820.5	7	61.4	24	81.5	24	10.1	7	28.4	7	28.7	7	17.7	29
		08	2523	2523	528.9	512	619.7	512	848.7	505	227.2	810	280.5	810	108.5	848	564.6	1916	571.0	1916	267.6	2101
		09	2844	2844	542.4	0	521.3	0	889.6	0	239.1	75	284.2	75	127.3	260	237.7	249	240.2	249	62.2	65
		10	2479	2479	501.2	382	562.5	382	922.3	134	208.5	525	246.2	525	121.6	710	342.0	719	345.3	719	68.4	180
100	15	01	2954	2954	1064.2	755	1095.5	755	1418.2	535	330.9	1843	402.2	1843	167.4	1229	193.9	951	196.0	951	153.7	1312
		02	2775	2775	976.8	3657	957.2	3657	1540.0	2132	294.7	3083	365.4	3083	121.3	1896	669.8	4060	676.6	4060	284.8	4050
		03	2617	2617	958.1	377	981.7	377	1488.3	501	223.8	243	268.6	243	69.6	163	273.9	262	277.3	262	69.6	260
		04	2817	2817	2445.2	13148	2477.4	13148	3378.2	15874	2572.3	23009	3135.7	23009	2335.1	22812	3459.7	32884	3491.2	32884	1816.5	13910
		05	2411	2411	791.4	2146	916.8	2146	1716.6	4531	369.0	4274	428.7	4274	254.8	3726	582.4	3605	588.0	3605	465.4	4668
		06	3879	3879	1381.9	1356	1334.3	1356	1892.5	3089	796.4	2849	915.4	2849	497.7	2333	356.5	2036	359.9	2036	433.5	3144
		07	2372	2372	811.6	5	718.3	5	1326.8	7	84.3	0	93.6	0	13.8	7	76.3	0	77.0	0	21.7	0
		08	3281	3281	1653.4	4842	1616.1	4842	2305.4	9845	1642.1	15929	1820.1	15929	1701.0	14575	2226.1	12751	2237.5	12751	1167.9	8923
		09	2993	2993	860.3	149	825.2	149	1328.6	83	155.6	35	172.1	35	38.0	214	148.1	163	149.2	163	57.3	586
		10	2544	2544	1036.0	1813	1017.3	1813	1328.0	1778	333.0	1289	369.9	1289	216.2	2224	596.9	2537	508.2	2537	273.3	2215
120	15	01	4065	4065	988.1	112	924.9	112	1133.3	107	928.9	60	1185.4	60	271.1	20	966.4	63	756.5	63	140.2	21
		02	3653	3653	521.4	0	550.5	0	847.1	0	99.7	0	123.5	0	70.3	66	61.3	20	47.6	20	18.3	0
		03	3756	3756	768.8	714	774.0	714	995.7	812	424.4	1757	527.7	1757	1128.0	707	119.1	329	98.2	329	243.2	538
		04	3211	3211	1465.1	521	1390.5	521	1700.4	824	1398.2	768	1652.4	768	1421.7	988	1247.5	752	1003.6	752	2324.8	3686
		05	4298	4298	tlim	4428	tlim	4079	4196.9	1551	1971.0	762	2303.1	762	4710.3	2429	3883.7	2397	3727.2	2397	3021.3	2655
		06	4512	4512	700.2	80	758.2	80	1965.8	539	1602.9	577	1853.1	577	2162.2	1449	1680.0	1239	1680.9	1239	294.1	78
		07	3463	3463	496.2	0	596.0	0	768.2	0	121.3	0	126.2	0	96.7	0	38.5	0	36.7	0	49.7	11
		08	3872	3872	1294.2	224	1202.8	224	1062.5	0	866.3	384	916.4	384	262.6	13	1613.5	475	1437.2	475	137.7	20

Continued on next page

Table A.12: (*Continued*)

					Generalized set partitioning						Time indexed						Arc-flow					
n	m	id	lb	ub	GSPP		GSPP_+		$\text{GSPP}_+^{\text{rc}}$		TI		TI_+		TI_+^{rc}		AF		AF_+		AF_+^{rc}	
					t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd
		09	4176	4176	1600.7	1132	1582.2	1132	2043.7	784	1059.4	254	1072.5	254	1103.4	726	1962.4	1606	1974.5	1606	1098.9	2095
		10	3880	3880	2843.8	7347	2938.3	7347	2735.1	4309	868.8	958	877.8	958	2839.9	6484	1536.7	1768	1550.1	1768	1864.5	6135
150	15	01	8219	8219	tlim	402	tlim	524	tlim	900	tlim	1035	tlim	1273	tlim	544	tlim	1798	tlim	2055	7003.8	3336
		02	6737.7	6742	tlim	462	tlim	550	tlim	920	tlim	527	tlim	521	tlim	499	tlim	1500	tlim	1497	tlim	2709
		03	4655	4655	858.0	0	922.4	0	1151.8	0	279.3	0	284.5	0	188.3	0	51.0	0	51.0	0	43.2	40
		04	7303	7303	tlim	3138	tlim	4252	5172.1	2128	tlim	2119	tlim	2052	tlim	620	tlim	493	tlim	487	tlim	1820
		05	6563	6563	3065.4	3854	2665.3	3854	tlim	8126	tlim	8698	tlim	8616	tlim	7297	4092.0	28055	3761.3	28055	2234.0	3629
		06	6348	6359	tlim	557	tlim	795	tlim	965	tlim	422	tlim	420	tlim	693	tlim	402	tlim	662	tlim	1664
		07	6343	6343	5145.4	1673	5503.4	1673	5432.6	1427	5076.3	1714	5174.9	1714	5689.4	1909	5088.5	1932	4914.7	1932	6983.2	1928
		08	7940	7940	tlim	85081	tlim	74811	5340.6	24680	tlim	4348	tlim	3871	tlim	63629	tlim	105292	tlim	103727	7011.9	45537
		09	8242	8242	3969.9	6257	4086.0	6257	2114.3	961	4265.1	10150	4815.8	10150	2507.5	1087	4510.6	11856	4533.8	11856	3390.9	10018
		10	6012.9	6016	tlim	3946	tlim	3559	tlim	5953	tlim	2637	tlim	2274	tlim	2874	tlim	2956	tlim	2980	tlim	3137
Avg.			-	-	1872.4	3182	1886.5	2993	2032.6	2019	1600.8	2154	1678.4	2137	1611.5	3163	1703.5	5047	1666.5	5024	1435.5	2949

Table A.13: Detailed results for the new benchmark instances

					Time-indexed				Arc-flow			
n	m	id	lb	ub	TI_+		TI_+^{rc}		AF_+		AF_+^{rc}	
					t(s)	nd	t(s)	nd	t(s)	nd	t(s)	nd
200	15	01	12604	12609	tlim	98	tlim	396	tlim	68	tlim	113
		02	10319	10319	6301.8	116	tlim	177	3620.8	114	1911.0	41
		03	11296	11355	tlim	47	tlim	318	tlim	108	tlim	218
		04	15441	15441	tlim	107	tlim	300	6880.6	343	3636.4	1499
		05	18166	18352	tlim	11	tlim	67	tlim	17	tlim	35
		06	16869	16869	6836.4	567	6454.6	577	1015.5	531	1575.9	224
		07	13025	13226	tlim	37	tlim	66	tlim	81	tlim	78
		08	14182	14259	tlim	7	tlim	9	tlim	48	tlim	58
		09	18118	18118	tlim	100	1911.5	105	3879.7	132	4951.6	657
		10	17102	17118	tlim	19	tlim	39	tlim	68	tlim	203
250	20	01	15633	15769	tlim	7	tlim	27	tlim	13	tlim	34
		02	15776	15915	tlim	5	tlim	20	tlim	24	tlim	55
		03	16519	16606	tlim	4	tlim	8	tlim	15	tlim	28
		04	16423	16481	tlim	17	tlim	46	tlim	14	tlim	33
		05	15661	15837	tlim	1	tlim	2	tlim	14	tlim	34
		06	20060	20060	tlim	0	tlim	4	5180.6	10	tlim	19
		07	14284	14362	tlim	0	tlim	25	tlim	21	tlim	38
		08	16305	16383	tlim	0	tlim	12	tlim	10	tlim	20
		09	15864	15917	tlim	0	tlim	2	tlim	16	tlim	26
		10	16283	16371	tlim	0	tlim	24	tlim	10	tlim	21
Avg.			-	-	7136.9	57	6898.3	111	6428.9	83	6363.7	172